# KCSAT-ML: Probing Reasoning Models with Nationwide-Cohort Human Difficulty

Anonymous ACL submission


## Abstract

Math reasoning benchmarks have proliferated, yet most lack a per-item difficulty signal grounded in actual human performance. We introduce **KCSAT-ML**, a decade (2014–2025) of Korean College Scholastic Ability Test (KCSAT; *Suneung*) mathematics: 664 problems with a 339-item core set carrying official per-item error rates from nationwide cohorts of hundreds of thousands of examinees. We pair the benchmark with *Difficulty-aligned Reasoning Gain* (DRG): a score-orthogonal metric that asks whether a model's mistakes concentrate on the items humans found hard, or on items humans found easy. Together they expose, across a wide range of VLMs (and LLMs via OCR), three patterns: (i) low-budget accuracy collapses on the high-human-error tail at every model size; (ii) test-time scaling (TTS) raises token use roughly linearly with cohort error rate, while accuracy gains follow a non-monotonic curve; (iii) within a single family, TTS flips between *anti-scaling* on the hardest items and *overthinking* on easier ones — two faces of the same alignment failure. On DRG, models with near-identical accuracy can sit at near-opposite values: one model gets wrong what humans also find hard, while another solves the hardest items yet fails on items humans find easy — a contrast that aggregate accuracy hides.

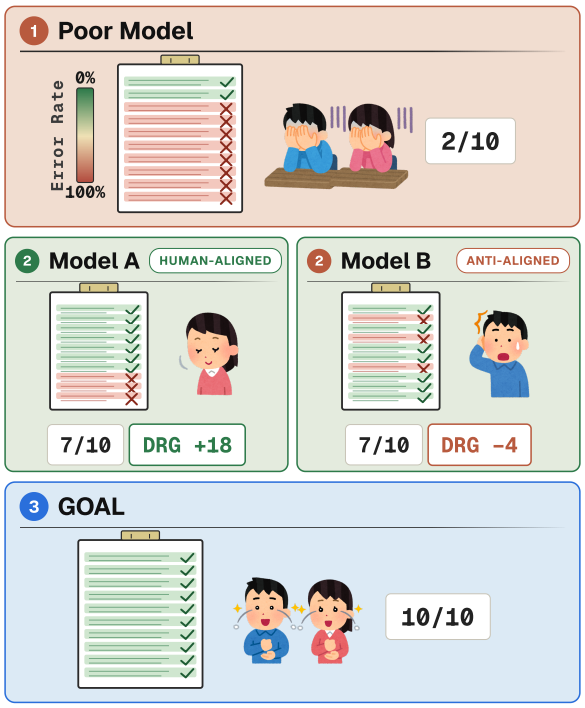




Figure 1: **Same score, opposite reasoning.** Two models both solve 7/10, but their mistakes land on opposite ends of the human-difficulty axis: Model A fails where humans also fail (DRG $+18$), Model B fails on items humans find easy (DRG $-4$). Accuracy alone cannot tell them apart; DRG (Sec. 4.7) can.

## 1 Introduction

Mathematical reasoning is now a central axis along which language and vision-language models are evaluated, and benchmarks across textual (Hendrycks et al., 2021), multimodal (Lu et al., 2023; Qiao et al., 2024), and olympiad-level (He et al., 2024; Gao et al., 2024) settings have multiplied accordingly. A benchmark, however, only resolves what its difficulty signal can resolve. Most existing math benchmarks report accuracy aggregated across items without any per-item notion of how hard each problem actually is, and the few that do attach difficulty labels rely on heuristic ordinal annotations (e.g., the 1–5 levels in MATH (Hendrycks et al., 2021)) assigned by the authors themselves. Without an item-level signal grounded in actual human performance, "difficulty" collapses into whatever the model finds hard. This conflates problem difficulty with model-specific weaknesses, and leaves the field without a measurement-based axis for comparing model behaviour.

KoNET (Park and Kim, 2025) takes an important step in this direction by attaching official examinee

statistics to items from the Korean College Scholastic Ability Test (KCSAT; *Suneung*), Korea's annual high-stakes college-entrance examination. Its mathematics subset, however, contains only 21 items with human error rates — too few to characterise the high-difficulty tail or support fine-grained per-item analysis. Moreover, the human signal is used primarily as additional reporting metadata, rather than as a tool for analysing how model behaviour varies with difficulty. The field therefore lacks both a sufficiently large mathematics benchmark with item-level human statistics and tools to turn such statistics into an analysis axis.

We introduce **KCSAT-ML**, a decade (2014–2025) of KCSAT mathematics: 664 problems in total, with a 339-item core set carrying official per-item error rates measured from nationwide cohorts of hundreds of thousands of examinees. Unlike author-assigned ordinal tiers, these rates are measured from examinee performance under standardised conditions, supplying a continuous, behaviourally grounded difficulty signal at a scale unavailable in prior math benchmarks.

This signal makes it possible to ask a question more demanding than aggregate accuracy: whether the items a model gets wrong are also the items humans find hard. A model that solves an olympiad-level problem but stumbles on a textbook-level one is high-scoring yet unlike a human reasoner — and arguably less trustworthy. To make this notion measurable, we introduce *Difficulty-aligned Reasoning Gain* (DRG): a score-orthogonal metric that captures whether additional inference-time compute aligns a model's failure pattern with the human difficulty axis, rather than merely clearing easy residual errors. We use this framework on KCSAT-ML to probe a question central to modern reasoning models: *when* the extra compute spent by techniques such as Chain-of-Thought prompting (Wei et al., 2022b; Wang et al., 2022) and *test-time scaling* (TTS) (Snell et al., 2024; Bi et al., 2024; Muennighoff et al., 2025; DeepSeek-AI, 2025) actually helps, and how its cost scales with the difficulty humans observe.

The pattern is sharp. Low-budget accuracy collapses on the high-human-error tail at every model size, and TTS only partially recovers it: token use grows with cohort error rate, but accuracy gains follow a non-monotonic curve. Within a single family, the same TTS turn-on yields opposite effects at the two extremes. On the hardest items, *anti-scaling* appears: a smaller w. TTS variant beats a larger wo. TTS sibling. On easier items, *overthinking* appears: extra reasoning derails an already-correct answer. Both are alignment failures — the model's competence and the human difficulty axis come apart, in opposite directions. DRG turns this into a model-level diagnostic: across our 22 model families, pairs with near-identical accuracy can sit at near-opposite DRG values, showing that final scores can hide near-opposite reasoning profiles.

Our contributions are as follows:

- **KCSAT-ML, a human-grounded difficulty instrument.** We release KCSAT-ML (2014–2025; 664 items, 339 with official cohort error rates from hundreds of thousands of examinees) and show that the human signal surfaces model behaviour that author-assigned ordinal labels miss.
- **DRG: a metric for human-aligned reasoning.** *Difficulty-aligned Reasoning Gain* (DRG) is a score-orthogonal summary of whether additional inference-time compute aligns a model's failure pattern with the human difficulty axis, rather than merely clearing easy residual errors. On DRG, models with near-identical accuracy can sit far apart, and that placement is largely fixed at the very first TTS turn-on.
- **A difficulty-conditioned view of TTS.** Using DRG and the cohort signal, we map TTS behaviour across the difficulty range: a difficulty-dependent collapse–recovery, near-linear token cost in cohort error rate, and two asymmetric failure modes at the extremes (*anti-scaling* on Hard, *overthinking* on Easy).

## 2 Related Work

**Mathematical reasoning benchmarks.** Existing math reasoning benchmarks span textual (Hendrycks et al., 2021), multimodal (Lu et al., 2023; Qiao et al., 2024), competition (He et al., 2024; Gao et al., 2024), and research-level mathematics. Soohak (Son et al., 2026), a 439-problem benchmark authored by 64 mathematicians, pushes evaluation past olympiad difficulty now that frontier models have reached IMO gold. Most of these benchmarks rely on author-assigned ordinal tiers, or provide no per-item difficulty signal at all. KoNET (Park and Kim, 2025) is closest in spirit to our work: it attaches nationwide-cohort statistics to Korean national exam items, but its

KCSAT mathematics subset contains only 21 items with human error rates, and the signal is reported as metadata rather than used as an analysis axis. KCSAT-ML extends this direction along two complementary axes — a 339-item core set with cohort error rates from hundreds of thousands of examinees, and DRG, which turns the signal into a probe of where model failures and TTS gains concentrate along the human-difficulty axis, and at what cost. Beyond a single label per item, each core item carries two independent signals: official point values (3 or 4) set by the exam designers, and item-level cohort error rates.

**Test-time scaling and inverse scaling.** Test-time scaling extends inference compute via chain-of-thought (Wei et al., 2022b), self-consistency (Wang et al., 2022), reasoning tokens (Snell et al., 2024; Muennighoff et al., 2025), process-level verification (Lightman et al., 2024), and adaptive routing (Liao et al., 2025; Zhang et al., 2025). Recent work has begun to document failure modes within this paradigm, including overthinking on simple problems in o1-like models (Chen et al., 2024). Sec. 4.5 situates such failures along the human-difficulty axis: anti-scaling on the hardest items and overthinking on the easiest are two faces of the same alignment failure, jointly summarised by DRG. The key distinction is the axis of failure: the inverse-scaling literature (McKenzie et al., 2023) treats parameter count as the sole failure-inducing axis, whereas anti-scaling is a two-axis effect: at the high-difficulty tail, a smaller model with inference-time compute can reverse the parameter-count gap against a larger low-budget sibling.

## 3 KCSAT-ML and Experimental Setup

### 3.1 KCSAT-ML

**KCSAT-ML** (KCSAT Mathematics Longitudinal) compiles a decade of Korean College Scholastic Ability Test (KCSAT) mathematics together with item-level error rates measured from hundreds of thousands of examinees. KCSAT items typically mix symbolic expressions with diagrams and graphs, so the benchmark is multimodal by default. KCSAT-ML extends the KCSAT subset of KoNET (Park and Kim, 2025), which contains 21 math items with human statistics, to 664 problems spanning 2014–2025, with official per-item error rates for a 339-item core subset. KCSAT mathematics has been administered under several variants (e.g., Type-A and Type-B tracks before 2022, and a common-plus-elective structure from 2022 onward); we aggregate all publicly released variants within this period. All evaluations and difficulty-conditioned analyses in this paper use the 339-item core set. The remaining items, generally easier problems for which item-level statistics are not publicly released, provide longitudinal coverage but are not used in our main experiments.

| | MathVista | KoNET Math | KCSAT-ML (Ours) |
|---|---|---|---|
| #Item (core / total) | 1,000 / 6,141 | 24 | 339 / 664 |
| Year | - | 2024 | 2014–2025 |
| Human Statistics | ✗ | ✓(21 items) | **✓(339 items)** |

Table 1: **Comparison with existing math benchmarks.** KCSAT-ML provides nationwide-cohort error rates on 339 items, expanding KoNET's KCSAT subset (21 with human statistics) by 16×.

**Construction.** We collect official PDF booklets released by the Korea Institute of Curriculum and Evaluation (KICE)[1] and parse them into item-level instances. Each item is preserved as a single image of the original exam layout, matched with structured metadata (year, form, question number, point value, answer type) and, where available, official human statistics. This image-first format keeps the model input close to the setting in which cohort error rates were measured: VLMs are evaluated directly on the image, while OCR[2] enables supplementary text-only LLM evaluation. We release annotations and scripts that reconstruct item images from the public KICE PDFs (details in Appendix A).

**Human statistics.** Each annual KCSAT yields item-level error rates from the nationwide cohort of roughly half a million examinees, providing an external, temporally consistent, continuous human-grounded difficulty signal. The official point values (3 or 4 in our core set) provide a complementary heuristic tier; Sec. 4.4 quantifies the gap between the two signals.

**Difficulty tiers.** Table 1 compares KCSAT-ML against existing math benchmarks. All 339 core items are 3- or 4-point questions[3] (4-point: 263; 3-point: 76), each typically requiring multi-step reasoning. Using official human error rates rather than point value alone, we bin the 339 core items

[1] https://www.kice.re.kr

[2] https://github.com/deepseek-ai/DeepSeek-OCR

[3] KCSAT mathematics items are scored at 2, 3, or 4 points; cohort error rates are publicly released only for the higher-point 3- and 4-point bands, which dominate the core set.

into three tiers: *Easy* (0–50%: 148 items), *Medium* (51–75%: 113 items), and *Hard* (76–100%: 78 items).

### 3.2 Models and TTS Conditions

We evaluate open-weight LLMs/VLMs and closed-source APIs on the KCSAT-ML core set. VLMs receive the problem image directly, while LLMs are provided OCR text. We report *Score* (accuracy, %) and *Avg.Tok* (average generated output tokens per item) in Table 2.

**Answer evaluation.** Core items are reformulated into a short-answer format to remove multiple-choice guessing while preserving original content and layout. Predictions are scored with an LLM-based equivalence judge (`GPT-5-2025-08-07`) that performs a narrow equivalence check between the predicted final answer and the reference (e.g., whether $\frac{1}{2}$ and $0.5$ denote the same value) and outputs `Correct`/`Incorrect`. The judge inspects the model's full response but evaluates only the equivalence of the extracted final answer against the fixed ground-truth, not the logical correctness of the reasoning chain; it is also blind to model identity and TTS configuration, so judge-induced bias across systems should be small. Full prompt templates appear in Figure 13.

**Inference settings (wo. TTS vs. w. TTS).** We compare (i) a low-budget, answer-only baseline (*wo. TTS*) against (ii) higher-budget settings (*w. TTS*) that allocate additional inference-time compute, either by adjusting inference settings of the same checkpoint or by using a reasoning-optimised variant where available (heterogeneity discussed in Limitations).

**Data contamination check.** KCSAT items are publicly released, raising a reasonable contamination concern. We anchor the check on GPT-5 (cut-off ∼September 2024, preceding the November 2024 administration of the 2025 exam): under memorisation we would expect a drop on post-cut-off items and a shrinking wo./w. TTS gap on pre-cut-off ones, but observe neither, and the same trajectory appears for Claude-Sonnet-4.5 and Gemini-3-Pro (Figure 2). Full evidence is detailed in the Limitations section.

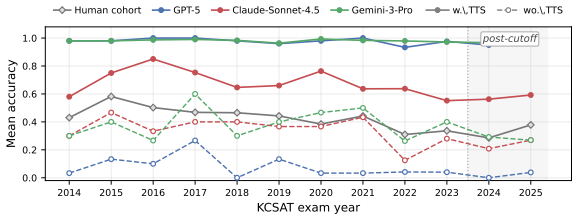



Figure 2: **Cross-model temporal accuracy on KCSAT-ML.** Per-year mean accuracy under wo. TTS (dashed) and w. TTS (solid) for GPT-5, Claude-Sonnet-4.5, and Gemini-3-Pro (2014–2025); grey diamonds: human cohort. Shaded: post-cutoff window anchored on GPT-5.

| Model | TTS | Avg.Tok | Score |
|---|---|---|---|
| ***Open-Weight LLMs (with OCR)*** | | | |
| Exaone-4.0-32B | | 6 | 3.8 |
| **Exaone-4.0-32B** | ✓ | **2,396** | **39.8** |
| Qwen3-30B-A3B-Instruct | | 5 | 7.4 |
| Qwen3-30B-A3B-Instruct | ✓ | 4,063 | 62.2 |
| **Qwen3-30B-A3B-Thinking** | ✓ | **7,678** | **74.0** |
| ***Open-Weight VLMs*** | | | |
| HyperCLOVA-X-Seed-Think-32B | | 746 | 39.5 |
| **HyperCLOVA-X-Seed-Think-32B** | ✓ | **7,431** | **87.2** |
| Qwen3-VL-8B-Instruct | | 58 | 6.5 |
| Qwen3-VL-8B-Instruct | ✓ | 5,450 | 46.3 |
| **Qwen3-VL-8B-Thinking** | ✓ | **8,842** | **52.2** |
| Qwen3-VL-32B-Instruct | | 5 | 6.5 |
| Qwen3-VL-32B-Instruct | ✓ | 3,703 | 54.6 |
| **Qwen3-VL-32B-Thinking** | ✓ | **7,808** | **61.7** |
| ***Closed-Source APIs (Selected Baselines)*** | | | |
| GPT-5.1 (None) | | 13 | 8.8 |
| **GPT-5.1 (High)** | ✓ | **6,587** | **93.2** |
| Claude-Opus-4.5 (Non-think) | | 7 | 13.3 |
| **Claude-Opus-4.5 (Think)** | ✓ | **3,617** | **87.6** |
| Gemini-3-Pro-Prev (Low) | | 23 | 37.4 |
| **Gemini-3-Pro-Prev (High)** | ✓ | **6,913** | **99.4** |

Table 2: **Selected KCSAT-ML Results.** Full results in Appendix Table 8. Baselines are shaded gray, their TTS counterparts blue.

| | MathVista | | | KCSAT-ML | | |
|---|---|---|---|---|---|---|
| **Model** | **wo. TTS** | **w. TTS** | **Δ** | **wo. TTS** | **w. TTS** | **Δ** |
| Gemini-2.5-Flash | 75.3 | 79.4 | +4.1 | 11.1 | 86.7 | +75.6 |
| GPT-5-Mini | 59.6 | 79.1 | +19.5 | 5.0 | 86.1 | +81.1 |
| Qwen3-VL-32B | 83.8 | 85.9 | +2.1 | 6.5 | 61.7 | +55.2 |

Table 3: **Effect of Test-Time Scaling (TTS).** wo. TTS: each model's default decoding; w. TTS: the official reasoning configuration of the same family. MathVista numbers are on *testmini*. The wo. TTS floor differs across benchmarks (high on MathVista, near zero on KCSAT-ML), which partly accounts for the Δ disparity.

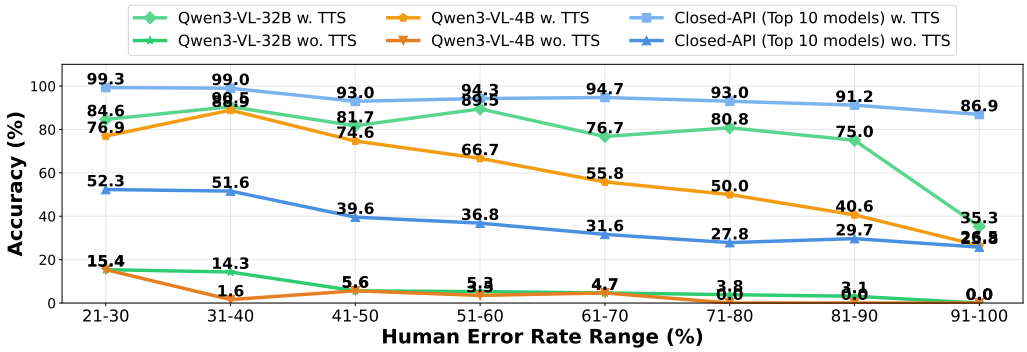



Figure 3: **Model scaling vs. test-time scaling (TTS) by human difficulty.** KCSAT-ML accuracy across human error-rate bins for Qwen3-VL (4B vs. 32B) and the top-10 closed APIs, with and without TTS.

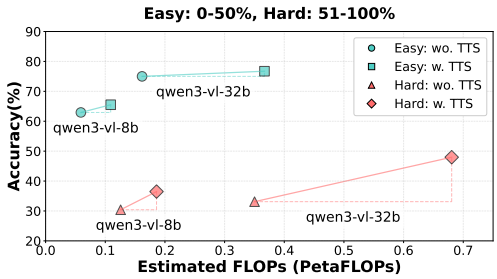



Figure 4: **When scaling helps: Qwen3-VL-8B vs. 32B.** Accuracy vs. estimated FLOPs on Easy and Hard subsets (split at 50% cohort error).

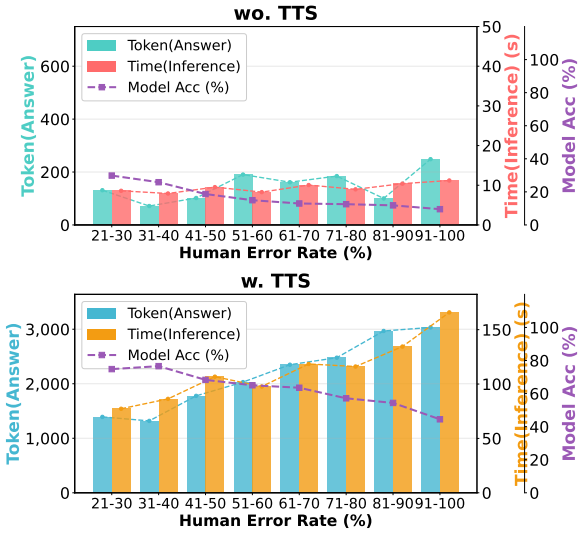



Figure 5: **Cost–accuracy trade-off of TTS.** Output tokens (blue), inference time (orange), and accuracy (purple) by human error rate, under wo. TTS (top) and w. TTS (bottom).

# 4 Analysis and Findings

## 4.1 Overall Impact of Test-Time Scaling

Table 2 reveals a clear *collapse–recovery* pattern: most models collapse to single-digit accuracy under low decoding budgets and recover to 60–99% once TTS is enabled (e.g., HyperCLOVA-X-Seed-Think-32B 39.5 → 87.2, Gemini-3-Pro-Preview reaches 99.4). The wo./w. TTS gap on KCSAT-ML far exceeds that on MathVista (Table 3), leaving substantial room for inference-time compute to matter and making KCSAT-ML a sharper instrument for studying when reasoning budgets are necessary.

## 4.2 Difficulty-Conditioned Scaling: Parameters vs. Test-Time Budget

Figure 3 shows where the recovery happens. Stratified by official human error-rate bins, accuracy falls off abruptly as human error rises under the low-budget setting (parameter scaling mainly helps the easier bins), while TTS reverses this pattern on the Hard tier ($\geqslant$ 76% human error), so an open-weight VLM with TTS (e.g., Qwen3-VL-32B) can outperform the average closed-source API under low budgets. Figure 4 drills further into Qwen3-VL 8B vs. 32B: on easier items, parameter scaling dominates and even aggressive TTS on the 8B model does not catch up; on harder items the picture inverts, with TTS producing a steeper gain than scaling 8B to 32B under *wo. TTS*. Beyond moderate difficulty, inference budget therefore dominates parameter scaling (Hard-tier ranking is robust across $k$-fold splits; Appendix C).

> ***Finding 1:*** Parameter scaling helps Easy items, while TTS dominates Hard ones: a 4× smaller w. TTS variant can beat a larger wo. TTS sibling on the hardest items.

## 4.3 Cost–Accuracy Trade-off

Figure 5 adds the cost axis. Under the low-budget setting, output length and latency are roughly constant across difficulty bins even as accuracy falls. Under TTS, output tokens grow roughly linearly

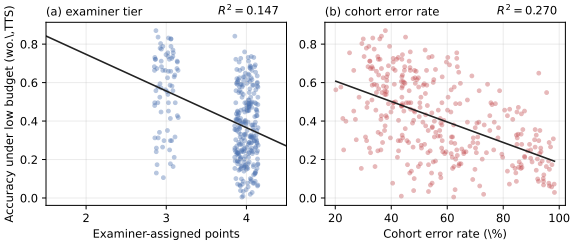



Figure 6: **Cohort error rate predicts model behaviour more strongly than examiner-assigned tiers.** Per-item mean accuracy (wo. TTS) vs. (a) examiner points and (b) cohort error rate; $N$=339.

with cohort error rate (Pearson $r$=0.59, $p$<$10^{-30}$; slope $\approx$ 102 tokens per pp of cohort error): models spend more compute on items humans find hardest. The accuracy gain, however, follows a non-monotonic curve, peaking at moderate difficulty (Sec. 4.4) and separating the cost of TTS from the accuracy benefit it delivers. Across closed-source models, monetary cost and accuracy are positively but imperfectly correlated ($r$=0.45, Spearman $\rho$=0.66); several mid-tier configurations with larger reasoning budgets reach both higher accuracy and lower cost than higher-tier models under low budgets, so budget allocation often outweighs model tier in practical cost-effectiveness (per-model dollar costs in Appendix B).

***Finding 2:*** TTS *cost* scales linearly with human difficulty ($r$=0.59), but TTS *gain* is non-monotonic, peaking at moderate (not maximal) difficulty.

### 4.4 Cohort Error Rate as a Stronger Difficulty Signal

Do KCSAT's *examiner-assigned point values*, a heuristic tier annotation similar to the 1–5 levels in MATH (Hendrycks et al., 2021), already carry the information that cohort error rates do? We compare the two by regressing per-item mean model accuracy on each signal in turn.

The cohort error rate is the stronger signal. It predicts per-item mean accuracy with nearly twice the explanatory power of examiner-assigned points ($R^2$ = 0.27 vs. 0.15 under wo. TTS; 0.18 vs. 0.05 under TTS), and a joint OLS regression confirms the cohort signal carries information the heuristic tier does not (full statistics in Appendix B). This continuous, behaviourally grounded variation is what surfaces the non-monotonic TTS-gain curve hidden under coarse point-value tiers.

***Finding 3:*** Human-grounded difficulty predicts model behaviour with roughly 2$\times$ the explanatory power of examiner-assigned tiers.

### 4.5 Two Faces of TTS: Anti-Scaling and Overthinking

The non-monotonic gain curve from Sec. 4.4 is not one effect: at the item level, it splits into two opposite failure modes at the difficulty extremes. We isolate them here using the per-item logs and provide a fuller catalog in Appendix D.

**Anti-scaling on the hardest items.** On the 23 items with cohort error $\geqslant$ 93%, Qwen3-VL-32B-Instruct in force-non-think mode is majority-incorrect on every one (0/23), terminating with $\leqslant$ 4 output tokens. Adding TTS to the same 32B backbone (Qwen3-VL-32B-Thinking) recovers 6/23 items; the 4$\times$ smaller Qwen3-VL-8B-Thinking under TTS recovers 9/23, with the two TTS-on configurations overlapping on 4 items and the rest split roughly evenly. The smaller w. TTS variant therefore solves items the larger wo. TTS sibling cannot, and the same outcome holds against Qwen3-30B-A3B-Instruct (force-non-think): the source of improvement is inference-time compute, not parameter count. Five representative items with per-trial outcomes appear in Appendix Table 6(a).

**Overthinking on easier items.** At cohort error rates below 60% (204 items: the Easy tier plus the lower portion of Medium), the converse pattern emerges *within a single backbone*: TTS degrades accuracy on items the wo. TTS variant already solves. Counting items on which the wo. TTS variant is majority-correct but the w. TTS variant is majority-incorrect: 11/204 (5.4%) for Qwen3-VL-32B, 7/204 (3.4%) for Qwen3-30B-A3B, and 4/204 (2.0%) for GPT-OSS-120B. Each rate is modest in isolation, but the same direction appears across three families with distinct architectures and training pipelines (per-configuration decoding noise is much smaller, Appendix C), so overthinking is a systematic TTS failure mode rather than a per-item or per-model artifact, echoing concurrent observations on o1-like models (Chen et al., 2024). Per-family lists with token counts and qualitative failure modes (e.g., abandoning textbook approaches for higher-machinery alternatives) appear in Appendix Table 6(b).

Together, these patterns show TTS is difficulty-asymmetric: extra compute can be decisive on Hard items but harmful on Easy ones.

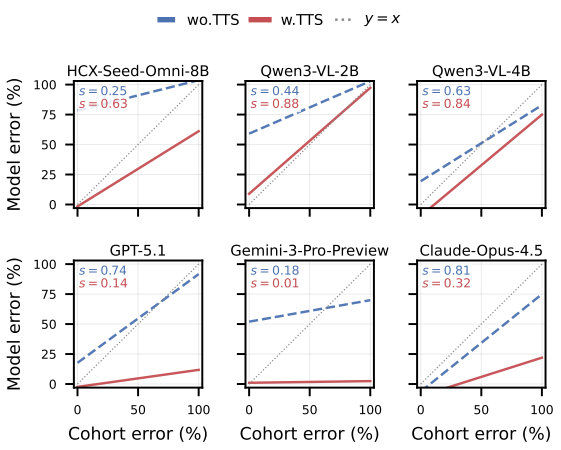



Figure 7: **Three regimes of TTS difficulty alignment.** Per-item model error rate ($y$) vs. cohort error rate ($x$), with wo. TTS (dashed) and w. TTS (solid) regression lines; $s$ marks the fitted slope (intercepts in Appendix B). Top: weak/mid-tier VLMs; bottom: frontier closed-source APIs.

***Finding 4:*** Within a single family, TTS reverses direction at the difficulty extremes: *anti-scaling* on Hard, *overthinking* on Easy. No uniform budget serves both.

### 4.6 TTS-Induced Difficulty Alignment Across Capability Levels

The TTS asymmetry from Section 4.5 appears at the item level within a single model. Viewed across models, TTS produces an analogous re-positioning along a single axis: the relationship between model error and human cohort error. The regression view compresses each model into two interpretable quantities. For each model with both wo. TTS and w. TTS runs we regress per-item model error rate on cohort error rate separately for each TTS condition, treating slope (failure–difficulty alignment) and intercept (offset relative to the $y{=}x$ diagonal) as the two summary statistics. The $y{=}x$ diagonal marks parity with the human cohort: the slope captures how closely model failures track human difficulty, the intercept measures vertical offset from human performance, and a regression line sitting below $y{=}x$ corresponds to error rates below the cohort across the difficulty range.

**Three regimes.** Figure 7 reveals three regimes (per-model slopes and intercepts are tabulated in Appendix B). *No-reasoning* (top row, wo. TTS): a near-flat regression sitting at high error, with failures uncorrelated with difficulty. *Human-aligned* (top row w. TTS, bottom row wo. TTS): slope $\approx 1$ with positive intercept, failures proportional to human difficulty. *Below-cohort* (bottom row w. TTS): slope $<0.4$ combined with a strongly negative intercept, so the model's error rate sits below the human cohort across the difficulty range. The same TTS turn-on drives opposite transitions depending on the starting point: it lifts weak models from *no-reasoning* into *human-aligned* (e.g., Qwen3-VL-2B's slope goes from 0.25 to 0.63), and frontier models from *human-aligned* into *below-cohort* (GPT-5.1's intercept drops to $-9.4$). This view places the anti-scaling and overthinking findings of Section 4.5 on common ground as item-level manifestations of one capability-conditional mechanism (Wei et al., 2022a): TTS reshapes *which items* a model fails on, with direction determined by the model's starting position.

***Finding 5:*** The same TTS turn-on drives *opposite* transitions for weak and frontier models, partitioning capability into three regimes: *no-reasoning*, *human-aligned*, *below-cohort*.

### 4.7 Beyond Accuracy: Measuring TTS Reasoning Quality

The patterns in Sections 4.5–4.6 indicate that TTS reshapes *which items* a model fails on, asymmetrically across difficulty. The question we now ask is one accuracy cannot answer: when TTS improves a model, does it reduce failures on the items humans found hardest, or does it mainly clean up easier residual errors?

**Sub-cohort fraction (SCF).** For a given (model, TTS setting) configuration and a difficulty bin $b \in \{\text{Easy}, \text{Hard}\}$, let $a_b$ be the number of items in bin $b$ where the model's error rate exceeds the human cohort's, and $\bar{a}_b$ the number where it falls below. We define the *sub-cohort fraction*

$$\mathrm{SCF}_b = \frac{a_b}{a_b + \bar{a}_b}.$$

$\mathrm{SCF}_b$ measures how often the model underperforms the human cohort within bin $b$. We use Easy ($\leqslant 50\%$ cohort error, 148 items) and Hard ($\geqslant 76\%$, 78 items) tiers as defined in Section 3.

**Difficulty-aligned Reasoning Gain (DRG).** Turning TTS on shifts the Easy/Hard SCF asymmetry, locating where the model's residual underperformance against humans concentrates. We define

$$\mathrm{DRG} = \left(\mathrm{SCF}_H^{\mathrm{off}} - \mathrm{SCF}_H^{\mathrm{on}}\right) - \left(\mathrm{SCF}_E^{\mathrm{off}} - \mathrm{SCF}_E^{\mathrm{on}}\right),$$

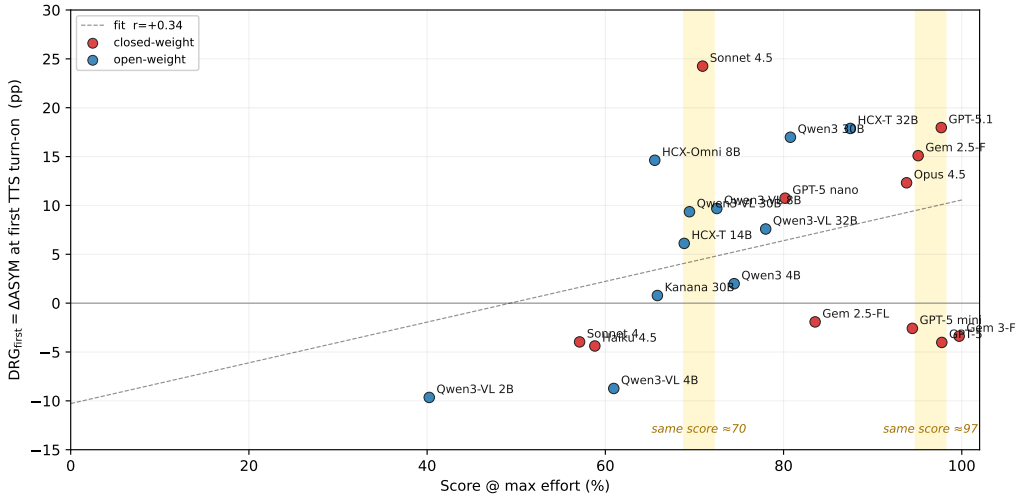



Figure 8: **DRG vs. accuracy.** For 22 model families with both wo. TTS and w. TTS runs, accuracy at max TTS effort ($x$) against DRG at first TTS turn-on ($y$); dashed: linear fit. Vertical gold bands highlight clusters with similar accuracy but wide DRG spread. Closed-weight (red) and open-weight (blue) distribute similarly along $y$ (Mann–Whitney $p$=0.95).

taken at the first TTS turn-on. Positive DRG means TTS preferentially closes the Hard-bin gap relative to humans; negative DRG means the gain is concentrated on Easy items while Hard failures remain; near-zero DRG means the improvement is roughly uniform across difficulty.

**(1) DRG is nearly orthogonal to accuracy.** Across the 22 families, the Pearson correlation between final score and DRG is only $r = +0.34$ (Figure 8). Within the high-accuracy cluster (max-effort score in $[86, 99]$), five frontier models span a 22 pp DRG range, and same-family generations can split sharply (e.g., Claude Sonnet-4 at $-4.0$ vs. Sonnet-4.5 at $+24.3$). The DRG sign decomposes the gain by difficulty tier: Sonnet-4.5's TTS budget preferentially clears above-cohort failures on the items humans found hardest, while Sonnet-4 concentrates its reduction on the Easy tier instead, so the two checkpoints reach comparable accuracy via near-opposite reasoning targets. Per-family values are listed in Appendix B.

**(2) DRG is locked in at the first TTS turn-on.** For families with multiple TTS budgets, DRG at the lowest TTS-on setting correlates with DRG at the highest setting at $r = +0.94$. Raising the budget lifts accuracy marginally but barely shifts DRG: TTS's effect on reasoning quality is locked in at turn-on.

**(3) Classification by DRG separates reasoning profiles.** Thresholding DRG at $\pm$ a small margin yields HIGH ($\geqslant +10$ pp; Hard-first saturation), MID ($-5$ to $+10$ pp), and LOW ($\leqslant -5$ pp; residual weaknesses concentrated in the Hard bin) groups (8/12/2 families). The LOW group shares a recurring failure: systematic over-engineering on textbook-level items, drawing thousands of reasoning tokens on problems wo. TTS solves in a handful (qualitative examples in Appendix B).

***Finding 6:*** DRG decouples *reasoning quality* from accuracy ($r$=0.34) and settles at the first TTS turn-on: same-score models can take opposite reasoning paths.

## 5 Conclusion

We introduce **KCSAT-ML**, a decade of KCSAT mathematics with nationwide-cohort human error rates, paired with *Difficulty-aligned Reasoning Gain* (DRG): a score-orthogonal metric for measuring how well a model's failures align with the human difficulty axis. Together, they show that test-time compute changes not only *how often* but *which items* models solve correctly: a difficulty-dependent collapse–recovery on the high-difficulty tail, *anti-scaling* and *overthinking* at opposite ends, and three capability-conditional alignment regimes across model families. The framework extends to any examination programme releasing item-level human statistics, offering a general probe of reasoning quality against human-grounded difficulty.

## Limitations

**Human signal as proxy.** Official human error rates are measured under the original KCSAT formats, while our evaluation reformulates items into a short-answer interface. The released rates therefore serve as a difficulty proxy rather than a calibrated measurement under our protocol, especially for originally multiple-choice items. Our analyses, however, depend on relative rankings across difficulty bins rather than absolute error rates, and the bin-level ordering is preserved under reformatting.

**LLM judge.** An LLM-based equivalence judge (`GPT-5-2025-08-07`; prompt in Appendix Figure 13) handles symbolic answers and formatting variation, but may introduce occasional judging errors. The judge is blind to model identity and TTS configuration, which limits systematic per-system bias, and we report trends that are stable across difficulty bins and inference budgets to further reduce sensitivity to individual judgments.

**Contamination.** KCSAT items are publicly released, but three observations argue against simple surface-form memorization as a driver of our results: (i) per-year wo. TTS accuracy shows no upward shift on pre-cutoff items relative to post-cutoff (Figure 2); (ii) wo. TTS collapse to single-digit accuracy on public items would not require reasoning budget to surface if final answers were merely recalled; and (iii) the year-invariant wo./w. TTS gap (per-family mean 34–91 pp) is incompatible with memorized completions surfacing without inference compute. More subtle exposure — such as memorized solution traces that surface under reasoning modes — remains possible, and evaluation on newly created or access-controlled items is a complementary direction rather than a prerequisite for the present conclusions.

**TTS heterogeneity across model families.** The wo./w. TTS contrast is not an identical intervention across model families: for some it changes only the inference-time budget or reasoning mode of a fixed checkpoint, for others it uses a reasoning-optimized checkpoint or vendor-provided reasoning mode. We treat this as a feature rather than a bug — it reflects how TTS is exposed in current model APIs and releases, and is the practical setting in which users choose inference budgets. Strictly causal same-checkpoint conclusions are restricted to model families where such ablations are available; the broader difficulty-conditioned patterns we report do not require them.

## References


Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. Forest-of-Thought: Scaling Test-Time Compute for Enhancing LLM Reasoning. In *International Conference on Machine Learning*.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024. Do Not Think That Much for 2+3=? On the Overthinking of o1-Like LLMs. *arXiv preprint arXiv:2412.21187*.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2024. Omni-MATH: A Universal Olympiad Level Mathematic Benchmark For Large Language Models. In *International Conference on Learning Representations*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Z. Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. OlympiadBench: A Challenging Benchmark for Promoting AGI with Olympiad-Level Bilingual Multimodal Scientific Problems. In *Annual Meeting of the Association for Computational Linguistics*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring Mathematical Problem Solving With the MATH Dataset. In *NeurIPS*.

Baohao Liao, Yuhui Xu, Hanze Dong, Junnan Li, C. Monz, Silvio Savarese, Doyen Sahoo, and Caiming Xiong. 2025. Reward-Guided Speculative Decoding for Efficient LLM Reasoning. In *International Conference on Machine Learning*.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's Verify Step by Step. In *International Conference on Learning Representations*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chun yue Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *International Conference on Learning Representations*.

Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, and 1 others. 2023. Inverse Scaling: When Bigger Isn't Better. *Transactions on Machine Learning Research (TMLR)*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Fei-Fei Li, Hanna Hajishirzi, Luke S. Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Sanghee Park and Geewook Kim. 2025. Evaluating multimodal generative AI with Korean educational standards. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 671–688, Albuquerque, New Mexico. Association for Computational Linguistics.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma Gongque, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, Runfeng Qiao, Yifan Zhang, Xiao Zong, Yida Xu, Muxi Diao, Zhimin Bao, Chen Li, and Honggang Zhang. 2024. We-Math: Does Your Large Multimodal Model Achieve Human-like Mathematical Reasoning? *arXiv preprint arXiv:2407.01284*.

C. Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *arXiv preprint arXiv:2408.03314*.

Guijin Son, Seungone Kim, Catherine Arnett, and 1 others. 2026. Soohak: A Mathematician-Curated Benchmark for Evaluating Research-level Math Capabilities of LLMs. *arXiv preprint arXiv:2605.09063*.

Xuezhi Wang, Jason Wei, D. Schuurmans, Quoc Le, Ed H. Chi, and Denny Zhou. 2022. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022a. Emergent Abilities of Large Language Models. *Transactions on Machine Learning Research*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Ed H. Chi, F. Xia, Quoc Le, and Denny Zhou. 2022b. Chain of Thought Prompting Elicits Reasoning in Large Language Models. In *Neural Information Processing Systems*.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. 2025. AdaptThink: Reasoning Models Can Learn When to Think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

## A Additional Results

This section provides additional qualitative examples, dataset statistics, full model results, and auxiliary analyses that support the main findings in Sec. 4.

**Qualitative examples.** Figures 9 and 10 present two representative items from the KCSAT-ML core set. These examples highlight two common sources of difficulty in KCSAT mathematics: (i) dense symbolic expressions embedded within the original exam layout, and (ii) diagram-dependent reasoning that requires multi-step geometric or algebraic manipulation. For readability, English translations are provided in the figures; however, all evaluations in this paper are performed using the original Korean problem statements.

**Dataset statistics and standardisation.** Table 7 summarises the composition of KCSAT-ML. The core subset ($N = 339$) is dominated by high-point items (4 points) and spans diverse answer types, including integers, fractions, and symbolic expressions. This diversity is why we score with an answer-equivalence judge rather than brittle string matching. To reduce multiple-choice guessing and keep the evaluation interface consistent, items with available human statistics are reformulated into a short-answer format while preserving the original exam content and layout.

**Full results across models and budgets.** Sec. 3 reports a selected subset of results in Table 2; Table 8 reports the complete results across all evaluated open-weight models and closed-source APIs. For several model families we include multiple budget configurations (e.g., vendor-provided budget tiers), giving a finer-grained view of the cost–accuracy trade-off than the binary wo./w. TTS setting alone.

**Positional analysis (sanity check for public items).** Because KCSAT items are publicly released, potential exposure to training data cannot be fully ruled out. Figure 11 compares human and model error rates by question number (i.e., position within an exam form) and does not show a clear monotonic trend that would suggest older items are systematically easier for the evaluated models. The temporal (year-by-year) view is given by the cross-model contamination check in Figure 2.

**Prompt templates and evaluation pipeline.** Figure 13 shows the exact prompt templates used throughout our experiments. The wo. TTS setting uses a low-budget, answer-only prompt; the w. TTS setting uses a higher-budget prompt that permits intermediate reasoning. The figure also includes the evaluation prompt used for LLM-based answer verification (Sec. 3), which is needed for robust scoring of symbolic answers.

**Structured metadata, annotation, and reconstruction.** Figure 14 shows an example of the structured metadata in KCSAT-ML, including item attributes (year, domain, question number, points), answer fields, and official human performance statistics. The figure also illustrates the image bounding boxes used to reconstruct problem images from publicly available KCSAT PDFs in the user environment, which makes the evaluation reproducible.

**Runtime estimation under a 100-minute exam budget.** Figure 12 addresses a practical question: if a model answers a full KCSAT mathematics form sequentially, would it finish within the official 100-minute limit (6000 s)? For each model, we estimate the total wall-clock time by summing tier-wise average inference latencies weighted by the expected number of questions in each tier. Tiers follow the Easy/Medium/Hard human error-rate bins defined in Sec. 3, augmented by a "Very Easy" bin that aggregates items without published cohort statistics (these items lie outside the 339-item core set and typically appear early in the form). A fixed system overhead for our evaluation pipeline (e.g., OCR/image handling and I/O) is then added. Models whose estimated total time is below 6000 s are marked as *pass*. This is an illustrative feasibility check under our protocol; absolute times may vary with hardware, batching, and API serving conditions.

## B Detailed Numerical Results

This appendix collects the per-model regression coefficients and DRG values referenced in Sections 4.4, 4.6, and 4.7.

**Cohort error rate vs. examiner points (Sec. 4.4).** Per-item mean model accuracy (averaged across all evaluated models) is regressed on each signal in turn. The cohort error rate yields $R^2 = 0.27$ (Pearson $r = -0.52$, $p<10^{-25}$) under wo. TTS, against $R^2 = 0.15$ for examiner-assigned points; under TTS the figures are $0.18$ and $0.05$. In a joint OLS regression with both signals as predictors, the

standardised coefficient on cohort error rate is $\beta = -0.44$, twice that of examiner points ($\beta = -0.22$), so the cohort signal carries independent variance that the heuristic tier alone does not capture.

**Per-model regression slopes/intercepts (Sec. 4.6).** For each model with both wo. TTS and w. TTS runs, we regress per-item model error rate ($y$, in %) on cohort error rate ($x$, in %) separately for each condition. Top-row models in Figure 7 (weak VLMs): HCX-Seed-Omni-8B (wo. TTS slope $0.25$, w. TTS slope $0.63$), Qwen3-VL-2B ($0.44 \rightarrow 0.63$), Qwen3-VL-4B ($0.63 \rightarrow 0.88$). Bottom-row models (frontier APIs): GPT-5.1 (wo. TTS slope $0.74$ / intercept $+12.6$, w. TTS slope $0.14$ / intercept $-9.4$), Claude-Opus-4.5 ($0.81/+8.7 \rightarrow 0.32/-9.4$). Gemini-3-Pro-Preview is a degenerate case under wo. TTS: `force-non-think` produces refusal-style outputs and yields mean accuracy $37\%$ with slope $0.18$; under w. TTS the model reaches $98\%$ accuracy with slope $0.01$ and intercept $-3.8$, placing it firmly in the below-cohort regime.

**Monetary cost per correct answer (Sec. 4.3).** Estimated API costs (USD) are computed from each provider's public per-token pricing. Among near-saturated closed-source systems, the cost per correct answer falls in a narrow band: GPT-5.1 \$0.064, Claude-Opus-4.5 \$0.073, Gemini-3-Pro-Preview \$0.081. Across closed-source models, several mid-tier configurations with larger reasoning budgets reach both higher accuracy and lower cost than higher-tier models under low budgets, so practical cost-effectiveness is governed more by budget allocation than by model tier.

**Per-family DRG values (Sec. 4.7).** Across the 22 model families with both TTS-off and TTS-on runs, DRG values cluster broadly between $-7$ and $+30$ pp. Within the high-accuracy cluster (max-effort score in $[86, 99]$), five frontier families span a 22 pp DRG range: GPT-5.1 ($+18.0$), Gemini-2.5-Flash ($+15.1$), GPT-5-mini ($-2.6$), Gemini-3-Flash-Preview ($-3.4$), and GPT-5 ($-4.0$). Same-family generations can split sharply too: Claude Sonnet-4 ($-4.0$) versus Sonnet-4.5 ($+24.3$). The DRG-by-budget correlation across multi-budget families is $r = +0.94$ between the lowest and highest TTS settings, so DRG saturates at the first TTS turn-on and barely shifts thereafter. Qualitative inspection of LOW-DRG and Easy-bin-failure models reveals systematic over-engineering: 2017 Math (Type-NA) #5 (3-point, $24\%$ cohort error) draws a $\sim$3,200-token solution from HCX-Think-32B invoking the Lambert $W$ function, and a textbook geometric-series item triggers $\sim$32,000 tokens of thinking trace in GPT-OSS-120B before producing an incorrect answer.

| Statistic of per-config Std (5 runs) | Value |
|---|---|
| N configurations | 76 |
| Mean Std (accuracy) | 0.037 |
| Median Std | 0.014 |
| Fraction with Std $< 0.05$ | 82.9% |

Table 4: **Decoding stability across 5 repeated runs.** Per-configuration standard deviation of overall accuracy on the 339-item core set, across 76 model$\times$budget configurations. Noise is small relative to the difficulty- and budget-conditioned effects in Sec. 4.

## C Robustness Analyses

Because the core set has 339 items and the Hard tier in particular contains 78 items, a natural concern is whether the difficulty-conditioned trends in Sec. 4 could be driven by decoding noise or by the limited size of the high-difficulty subset. We address both factors here.

**Per-model stability across repeated decoding.** We re-ran a representative subset of 76 model$\times$budget configurations five times each on the 339 core items, keeping all other settings identical, and computed the per-configuration standard deviation of overall accuracy across runs. Table 4 summarizes the distribution: the median standard deviation is $0.014$ and the mean is $0.037$ (in accuracy units), with $82.9\%$ of configurations exhibiting Std $< 0.05$. The decoding noise on aggregated accuracy is therefore much smaller than the differences between difficulty tiers ($\geqslant 25$ percentage points) and between wo./w. TTS settings ($\geqslant 30$ percentage points for most models) discussed in the main paper.

**Subset stability via $k$-fold splits on the Hard tier.** To check whether the size of the Hard tier (78 items) is sufficient for the model-ranking conclusions drawn in Sec. 4.4 and Figure 3, we partition the Hard tier into $k$ disjoint folds, compute per-fold model accuracy, and correlate the resulting per-fold ranking with the ranking obtained on the full Hard tier. Table 5 reports the Pearson correlation; even for $k$=10 (each fold contains only 7–8 items), per-fold model rankings remain almost perfectly aligned with the full-Hard-tier ranking

| $k$ (folds) | Items per fold | Pearson $r$ vs. full |
|---|---|---|
| 2 | 39 | 0.997 |
| 3 | 26 | 0.996 |
| 5 | 15–16 | 0.990 |
| 10 | 7–8 | 0.975 |

Table 5: **Subset stability of model rankings on the Hard tier.** For each $k$, the 78 Hard items are split into $k$ folds; per-fold rankings are correlated (Pearson) with the full-tier ranking and averaged. Rankings remain near-identical even for $k$=10.

($r = 0.975$). At $k$=5 (15–16 items per fold), correlation rises to $0.990$. This indicates that the relative ordering of models on Hard items is stable across random subsets and is not an artifact of the specific 78-item composition.

Together, these analyses indicate that (i) single-run aggregate accuracies are reliable proxies for repeated-run means, and (ii) the qualitative ranking of models on the Hard tier is robust to the specific subset of items used. The difficulty-conditioned conclusions in Sec. 4 therefore do not hinge on the limited size of the 78-item Hard subset.

## D Edge-Case Item Studies

Aggregate trends aside, KCSAT-ML's per-item human statistics let us isolate individual problems on which the difficulty-conditioned pattern is concretely visible. We pick two complementary edge cases from the 5-run logs of Sec. C: an extreme-difficulty item where a $4\times$ smaller model with TTS clearly beats a larger sibling under low-budget decoding (anti-scaling), and an Easy item where the same strong model's TTS variant degrades the wo. TTS answer (overthinking).

**Anti-scaling on extreme-difficulty items.** On 2022 Mathematics Probability & Statistics #30 (cohort error rate $96.4\%$, the upper extreme of the Hard tier; reference answer 191), Qwen3-VL-32B-Instruct under a strict low-budget setting (force-non-think) produces a 3-token answer "13" and is wrong (✗); the same backbone never recovers under repeated trials in this configuration. In contrast, Qwen3-VL-8B-Thinking with TTS is correct in every one of the five repeated runs (✓), converging to 191 via $\sim$9,600 structured reasoning tokens per trial. The smaller backbone with TTS therefore solves a problem on which a $4\times$ larger sibling under low-budget decoding fails completely.

The pattern is not item-specific. The upper block of Table 6 lists five Hard items (cohort error rates 93–98$\%$) on which the anti-scaling outcome holds: a $4\times$ smaller w. TTS configuration solves the item near-perfectly while the larger wo. TTS configuration outputs a $\leqslant$ 4-token answer and is wrong. The same effect appears against an even larger 30B mixture-of-experts baseline (Qwen3-30B-A3B-Instruct, force-non-think) on the identical item set, again pointing to inference-time compute rather than parameter count as the source of improvement.

| Item (KCSAT) | Cohort err. | wo. TTS (maj. / avg tok) | w. TTS (maj. / avg tok) |
|---|---|---|---|
| *(a) Anti-scaling: Qwen3-VL-32B-Instruct (force-non-think) wo. TTS vs. Qwen3-VL-8B-Thinking w. TTS* | | | |
| 2022 Prob. & Stat. #30 | 96.4% | ✗ / 3 tok | ✓ / ∼9,600 tok |
| 2020 Math NA #30 | 97.9% | ✗ / 2 tok | ✓ / ∼11,400 tok |
| 2018 Math NA #30 | 93.4% | ✗ / 3 tok | ✓ / ∼15,800 tok |
| 2020 Math GA #30 | 95.5% | ✗ / 3 tok | ✓ / ∼5,500 tok |
| 2024 Math Calc. #29 | 93.2% | ✗ / 3 tok | ✓ / ∼10,900 tok |
| *(b) Overthinking: same-backbone Instruct wo. TTS vs. Thinking w. TTS* | | | |
| 2019 Math NA #26 *[Qwen3-VL-32B]* | 46.8% | ✓ / 945 tok | ✗ / ∼2,900 tok |
| 2025 Math Calc. #26 *[Qwen3-VL-32B]* | 32.2% | ✓ / ∼1,000 tok | ✗ / ∼3,400 tok |
| 2024 Math Geo. #27 *[Qwen3-VL-32B]* | 58.2% | ✓ / ∼3,600 tok | ✗ / ∼14,800 tok |
| 2014 Math A #18 *[Qwen3-30B-A3B]* | 43.8% | ✓ / 718 tok | ✗ / ∼2,300 tok |
| 2019 Math GA #24 *[GPT-OSS-120B]* | 37.7% | ✓ / 576 tok | ✗ / ∼3,400 tok |

Table 6: **Edge-case items showing two faces of difficulty-conditional TTS.** Cells report the majority outcome across repeated trials (✓ = correct majority, ✗ otherwise) and average output tokens. (a) Anti-scaling: a smaller w. TTS variant solves Hard items that a larger wo. TTS baseline fails. (b) Overthinking: on easier items, the same-backbone Thinking variant fails despite spending more tokens. Model family varies per row in (b), shown in italics.

**Overthinking on Easy/Medium items.** The converse pattern appears on items well within model competence under direct answering. The lower block of Table 6 lists five items spanning 32–58$\%$ cohort error in which the *same-backbone* Instruct (wo. TTS) variant solves the problem in every trial while the Thinking (w. TTS) variant fails the majority of trials despite spending 3–6$\times$ more output tokens. The effect spans multiple families. For instance, on 2019 Mathematics (Type-NA) #26 (cohort error rate $46.8\%$), Qwen3-VL-32B-Instruct is correct in all five runs (✓) with $\sim$945 output tokens per trial, whereas Qwen3-VL-32B-Thinking expends $\sim$2,900 tokens per trial and is wrong in every run (✗). As a more striking instance, on 2019 Mathematics (Type-GA) #24 (cohort error rate $37.7\%$; reference answer 4), GPT-OSS-120B under the low-budget setting answers "4" in 576 output tokens (✓), but enabling TTS on the same backbone produces 3,000–4,000-token derivations that miscalculate to $\boxed{4\sqrt{13}}$ or $\boxed{16}$ in all five trials (✗).

Together, blocks (a) and (b) show that the same model family can shift from anti-scaling

| Statistic | KCSAT-ML (Full) | KCSAT-ML (Core) |
|---|---|---|
| Images | 664 | 339 |
| Questions | 664 | 339 |
| *Question Type* | | |
| Multiple-choice | 468 (70.5%) | – |
| Short-answer | 196 (29.5%) | 339 (100.0%) |
| Subjects | Mathematics (100%) | Mathematics (100%) |
| *Answer Type* | | |
| Integer | 450 (67.8%) | 227 (67.0%) |
| Fraction | 142 (21.4%) | 66 (19.5%) |
| Expression | 37 (5.6%) | 18 (5.3%) |
| Options | 18 (2.7%) | 17 (5.0%) |
| Decimal | 17 (2.6%) | 11 (3.2%) |
| *Points* | | |
| 2 points | 68 (10.2%) | 0 (0.0%) |
| 3 points | 313 (47.1%) | 76 (22.4%) |
| 4 points | 283 (42.6%) | 263 (77.6%) |
| *Human Error Rate* | | |
| None | 325 (48.9%) | 0 (0.0%) |
| Easy (0–50%) | 148 (22.3%) | 148 (43.7%) |
| Medium (51–75%) | 113 (17.0%) | 113 (33.3%) |
| Hard (76–100%) | 78 (11.7%) | 78 (23.0%) |

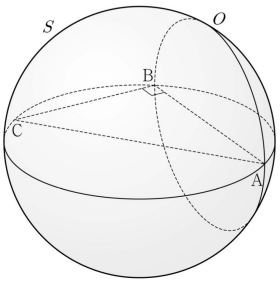
Table 7: **Statistics of KCSAT-ML.** Full benchmark (664 items) and the core subset (339 short-answer items with official human error rates) used in all difficulty-conditioned analyses.

(small+TTS $>$ big wo. TTS) to overthinking (wo. TTS $>$ w. TTS) within one benchmark, purely as a function of item difficulty.

## Example 1: 2025 KCSAT Geometry, Problem 28

**28.** 좌표공간에 $\overline{AB}=8$, $\overline{BC}=6$, $\angle ABC=\frac{\pi}{2}$ 인 직각삼각형 ABC와 선분 AC를 지름으로 하는 구 $S$가 있다. 직선 AB를 포함하고 평면 ABC에 수직인 평면이 구 $S$와 만나서 생기는 원을 $O$라 하자. 원 $O$ 위의 점 중에서 직선 AC까지의 거리가 4인 서로 다른 두 점을 P, Q라 할 때, 선분 PQ의 길이는? [4점]

① $\sqrt{43}$ ② $\sqrt{47}$ ③ $\sqrt{51}$ ④ $\sqrt{55}$ ⑤ $\sqrt{59}$




==========

**# Translate (in english)**
28. In a right triangle ABC on the coordinate plane,
AB = 8, BC = 6, and ∠ABC = $\pi$/2.

Let S be the sphere having segment AC as its diameter.
Consider the plane that contains line AB and is perpendicular to the plane ABC.
This plane intersects the sphere S in a circle, which we denote by O.

Among the points on circle O, let P and Q be two distinct points whose distances
from the line AC are both equal to 4.
Find the length of segment PQ.

[4 points]

**# Ground Truth**
$\sqrt{55}$

==========

**# Prompt**
Please solve the problem given in the image. IMPORTANT: Provide ONLY the final answer. Do NOT show your work, reasoning, thinking process, step-by-step solution, or any explanation. Just give the answer directly.

**# Response**
$2\sqrt{7}$

==========

**# Prompt**
Please solve the problem given in the image.

**# Response**
Let's solve the problem step-by-step using a coordinate system.

**1. Set up the coordinate system and define triangle ABC:**
Given $\overline{AB}=8$, $\overline{BC}=6$, and $\angle ABC=\frac{\pi}{2}$.
Let $B$ be the origin $(0,0,0)$.
Since $\angle ABC = 90^\circ$, we can place $A$ along the x-axis and $C$ along the y-axis.
So, $A=(8,0,0)$ and $C=(0,6,0)$.
The plane containing triangle $ABC$ is the $xy$-plane, which has the equation $z=0$.

...

The final answer is $\boxed{\sqrt{55}}$.

Response Model : Gemini 2.5 Flash

Figure 9: **Example of a geometry-based KCSAT-ML item.** A representative geometry problem (2025, #28) with sample wo. TTS and w. TTS responses from Gemini-2.5-Flash.

## Example 2: 2025 KCSAT Common, Problem 6

**6.** $\cos\left(\frac{\pi}{2}+\theta\right) = -\frac{1}{5}$ 일 때, $\frac{\sin\theta}{1-\cos^2\theta}$ 의 값은? [3점]

① $-5$ ② $-\sqrt{5}$ ③ $0$ ④ $\sqrt{5}$ ⑤ $5$

==========

**# Translate (in english)**
6. If cos(π/2 + θ) = -1/5, what is the value of sinθ / (1 − cos²θ)? [3 points]
① -5 ② -√5 ③ 0 ④ √5 ⑤ 5"

**# Ground Truth**
$5$

==========

**# Prompt**
Please solve the problem given above. IMPORTANT: Provide ONLY the final answer. Do NOT show your work, reasoning, thinking process, step-by-step solution, or any explanation. Just give the answer directly.

**# Response**
$\frac{3}{4}$

==========

**# Prompt**
Please solve the problem given above.

**# Response**
Given the problem: If \(\cos\left(\frac{\pi}{2} + \theta\right) = -\frac{1}{5}\), find the value of \(\frac{\sin\theta}{1 - \cos^2\theta}\).

1. **Using the co-function identity**:
   \[
   \cos\left(\frac{\pi}{2} + \theta\right) = -\sin\theta
   \]
   Given \(\cos\left(\frac{\pi}{2} + \theta\right) = -\frac{1}{5}\), we have:
   \[
   -\sin\theta = -\frac{1}{5} \implies \sin\theta = \frac{1}{5}
   \]

2. **Simplifying the expression**:
   The expression to evaluate is \(\frac{\sin\theta}{1 - \cos^2\theta}\). Using the Pythagorean identity \(1 - \cos^2\theta = \sin^2\theta\), we get:
   \[
   \frac{\sin\theta}{1 - \cos^2\theta} = \frac{\sin\theta}{\sin^2\theta} = \frac{1}{\sin\theta}
   \]

3. **Substituting the value of \(\sin\theta\)**:
   \[
   \frac{1}{\sin\theta} = \frac{1}{\frac{1}{5}} = 5
   \]

Thus, the value of \(\frac{\sin\theta}{1 - \cos^2\theta}\) is \(\boxed{5}\).

Response Model : HyperCLOVAX-SEED-Think-14B

Figure 10: **Example of an algebraic KCSAT-ML item.** A representative algebra problem (2025, #6) with sample wo. TTS and w. TTS responses from HyperCLOVAX-SEED-Think-14B.

### Open-Weight Models

| Model | TTS | Avg.Tok | Score |
|---|---|---|---|
| ***Open-Weight LLMs (Korean & Global)*** | | | |
| Solar-10-7B-Instruct-v1.0 | | 85 | 3.2 |
| **Solar-10-7B-Instruct-v1.0** | ✓ | **700** | **0.0** |
| Solar-Pro-Preview-Instruct-22B | | 5 | 2.4 |
| **Solar-Pro-Preview-Instruct-22B** | ✓ | **686** | **0.0** |
| Midm-2.0-Base-Instruct-12B | | 11 | 3.8 |
| **Midm-2.0-Base-Instruct-12B** | ✓ | **695** | **8.6** |
| SKT-AX-4.0-Light-7B | | 465 | 22.7 |
| **SKT-AX-4.0-Light-7B** | ✓ | **1,002** | **14.2** |
| SKT-AX-4.0-72B | | 9 | 5.3 |
| **SKT-AX-4.0-72B** | ✓ | **1,873** | **20.1** |
| GPT-OSS-20B | | 2,272 | 67.8 |
| **GPT-OSS-20B** | ✓ | **4,283** | **58.4** |
| Exaone-4.0-32B | | 6 | 3.8 |
| **Exaone-4.0-32B** | ✓ | **2,396** | **39.8** |
| Kanana-2-30B-A3B-Instruct | | 9 | 5.6 |
| **Kanana-2-30B-A3B-Instruct** | ✓ | **1,324** | **27.1** |
| **Kanana-2-30B-A3B-Thinking** | ✓ | **9,875** | **45.4** |
| HyperCLOVA-X-Seed-14B | | 55 | 6.8 |
| **HyperCLOVA-X-Seed-14B(Non-think)** | ✓ | **990** | **8.8** |
| **HyperCLOVA-X-Seed-14B(Think)** | ✓ | **8,646** | **53.7** |
| Qwen3-4B-Instruct | | 7 | 5.9 |
| **Qwen3-4B-Instruct** | ✓ | **4,274** | **49.6** |
| **Qwen3-4B-Thinking** | ✓ | **8,558** | **64.3** |
| GPT-OSS-120B | | 1,839 | 74.9 |
| **GPT-OSS-120B** | ✓ | **3,552** | **67.6** |
| Qwen3-30B-A3B-Instruct | | 5 | 7.4 |
| **Qwen3-30B-A3B-Instruct** | ✓ | **4,063** | **62.2** |
| **Qwen3-30B-A3B-Thinking** | ✓ | **7,678** | **74.0** |
| ***Open-Weight VLMs*** | | | |
| HyperCLOVA-X-Vision-3B | | 9 | 1.5 |
| **HyperCLOVA-X-Vision-3B** | ✓ | **749** | **4.4** |
| HyperCLOVA-X-Seed-Think-32B(Non-think) | | 746 | 39.5 |
| **HyperCLOVA-X-Seed-Think-32B(Think)** | ✓ | **7,431** | **87.2** |
| Qwen3-VL-2B-Instruct | | 821 | 1.2 |
| **Qwen3-VL-2B-Instruct** | ✓ | **10,650** | **5.9** |
| **Qwen3-VL-2B-Thinking** | ✓ | **12,379** | **18.3** |
| Qwen3-VL-4B-Instruct | | 112 | 3.2 |
| **Qwen3-VL-4B-Instruct** | ✓ | **6,668** | **28.3** |
| **Qwen3-VL-4B-Thinking** | ✓ | **9,975** | **35.4** |
| Qwen3-VL-30B-A3B-Instruct | | 92 | 7.7 |
| **Qwen3-VL-30B-A3B-Instruct** | ✓ | **5,699** | **35.4** |
| **Qwen3-VL-30B-A3B-Thinking** | ✓ | **7,602** | **45.4** |
| Qwen3-VL-8B-Instruct | | 58 | 6.5 |
| **Qwen3-VL-8B-Instruct** | ✓ | **5,450** | **46.3** |
| **Qwen3-VL-8B-Thinking** | ✓ | **8,842** | **52.2** |
| Qwen3-VL-32B-Instruct | | 5 | 6.5 |
| **Qwen3-VL-32B-Instruct** | ✓ | **3,703** | **54.6** |
| **Qwen3-VL-32B-Thinking** | ✓ | **7,808** | **61.7** |

### Closed-Source APIs

| Model | TTS | Avg.Tok | Score |
|---|---|---|---|
| GPT-5-Nano (Minimal) | | 10 | 1.2 |
| **GPT-5-Nano (Minimal)** | ✓ | **657** | **2.7** |
| **GPT-5-Nano (low)** | ✓ | **1,697** | **29.8** |
| **GPT-5-Nano (Medium)** | ✓ | **6,446** | **54.0** |
| **GPT-5-Nano (High)** | ✓ | **13,079** | **60.8** |
| o4-Mini (Low) | | 9 | 73.2 |
| **o4-Mini (Low)** | ✓ | **1,674** | **59.9** |
| **o4-Mini (Medium)** | ✓ | **2,976** | **74.3** |
| **o4-Mini (High)** | ✓ | **5,350** | **83.5** |
| GPT-5-Mini (Minimal) | | 9 | 5.0 |
| **GPT-5-Mini (Minimal)** | ✓ | **957** | **21.8** |
| **GPT-5-Mini (Low)** | ✓ | **1,495** | **62.2** |
| **GPT-5-Mini (Medium)** | ✓ | **2,741** | **76.7** |
| **GPT-5-Mini (High)** | ✓ | **5,104** | **86.1** |
| GPT-5 (Minimal) | | 8 | 7.7 |
| **GPT-5 (Minimal)** | ✓ | **538** | **17.1** |
| **GPT-5 (Low)** | ✓ | **2,000** | **82.9** |
| **GPT-5 (Medium)** | ✓ | **3,808** | **91.2** |
| **GPT-5 (High)** | ✓ | **6,442** | **94.1** |
| GPT-5.1 (None) | | 13 | 8.8 |
| **GPT-5.1 (None)** | ✓ | **691** | **15.0** |
| **GPT-5.1 (Low)** | ✓ | **1,916** | **73.7** |
| **GPT-5.1 (Medium)** | ✓ | **3,933** | **90.0** |
| **GPT-5.1 (High)** | ✓ | **6,587** | **93.2** |
| Claude-Sonnet-4 (Non-think) | | 328 | 15.6 |
| **Claude-Sonnet-4** | ✓ | **827** | **16.5** |
| **Claude-Sonnet-4 (Lite-think)** | ✓ | **2,201** | **23.9** |
| **Claude-Sonnet-4 (Think)** | ✓ | **3,594** | **31.9** |
| Claude-Sonnet-4.5 (Non-think) | | 549 | 33.3 |
| **Claude-Sonnet-4.5** | ✓ | **868** | **19.8** |
| **Claude-Sonnet-4.5 (Lite-think)** | ✓ | **2,524** | **41.6** |
| **Claude-Sonnet-4.5 (Think)** | ✓ | **3,732** | **49.3** |
| Claude-Haiku-4.5 (Non-think) | | 66 | 7.7 |
| **Claude-Haiku-4.5** | ✓ | **833** | **13.9** |
| **Claude-Haiku-4.5 (Lite-think)** | ✓ | **2,824** | **25.7** |
| **Claude-Haiku-4.5 (Think)** | ✓ | **4,440** | **33.3** |
| Claude-Opus-4.5 (Non-think) | | 7 | 13.3 |
| **Claude-Opus-4.5** | ✓ | **736** | **44.5** |
| **Claude-Opus-4.5 (Lite-think)** | ✓ | **2,477** | **79.1** |
| **Claude-Opus-4.5 (Think)** | ✓ | **3,617** | **87.6** |
| Gemini-2.5-Flash-Lite | | 3 | 6.9 |
| **Gemini-2.5-Flash-Lite** | ✓ | **4,850** | **8.8** |
| **Gemini-2.5-Flash-Lite (Think)** | ✓ | **11,933** | **66.7** |
| Gemini-2.5-Flash (Non-think) | | 4 | 11.1 |
| **Gemini-2.5-Flash** | ✓ | **3,218** | **17.7** |
| **Gemini-2.5-Flash (Think)** | ✓ | **7,595** | **86.7** |
| Gemini-2.5-Pro (Think) | | 3 | 96.1 |
| **Gemini-2.5-Pro (Think)** | ✓ | **6,886** | **93.2** |
| Gemini-3-Flash-Prev (Minimal) | | 1,028 | 76.3 |
| **Gemini-3-Flash-Prev (Minimal)** | ✓ | **1,008** | **62.5** |
| **Gemini-3-Flash-Prev (Low)** | ✓ | **3,317** | **88.2** |
| **Gemini-3-Flash-Prev (Medium)** | ✓ | **7,362** | **96.2** |
| **Gemini-3-Flash-Prev (High)** | ✓ | **6,210** | **98.5** |
| Gemini-3-Pro-Prev (Low) | | 23 | 37.4 |
| **Gemini-3-Pro-Prev (Low)** | ✓ | **2,345** | **83.8** |
| **Gemini-3-Pro-Prev (High)** | ✓ | **6,913** | **99.4** |

Table 8: **Full results on KCSAT-ML.** Complete results across all models and settings; extends Table 2. Baselines are shaded gray, their TTS counterparts blue.

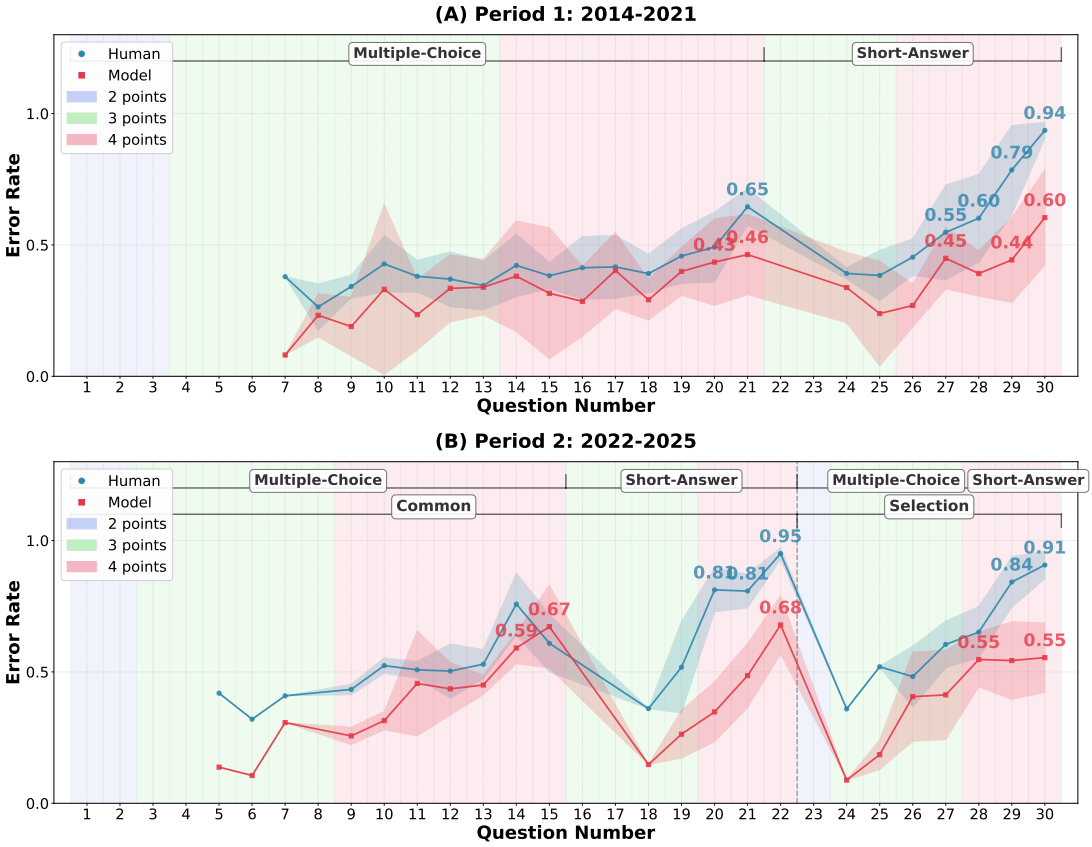



Figure 11: **Human vs. model error rates across question numbers.** Error rates over 30 questions in KCSAT-ML, grouped by question format (multiple-choice vs. short-answer) and period: (A) 2014–2021 and (B) 2022–2025. Periods are split by the 2022 introduction of the common-plus-elective structure.

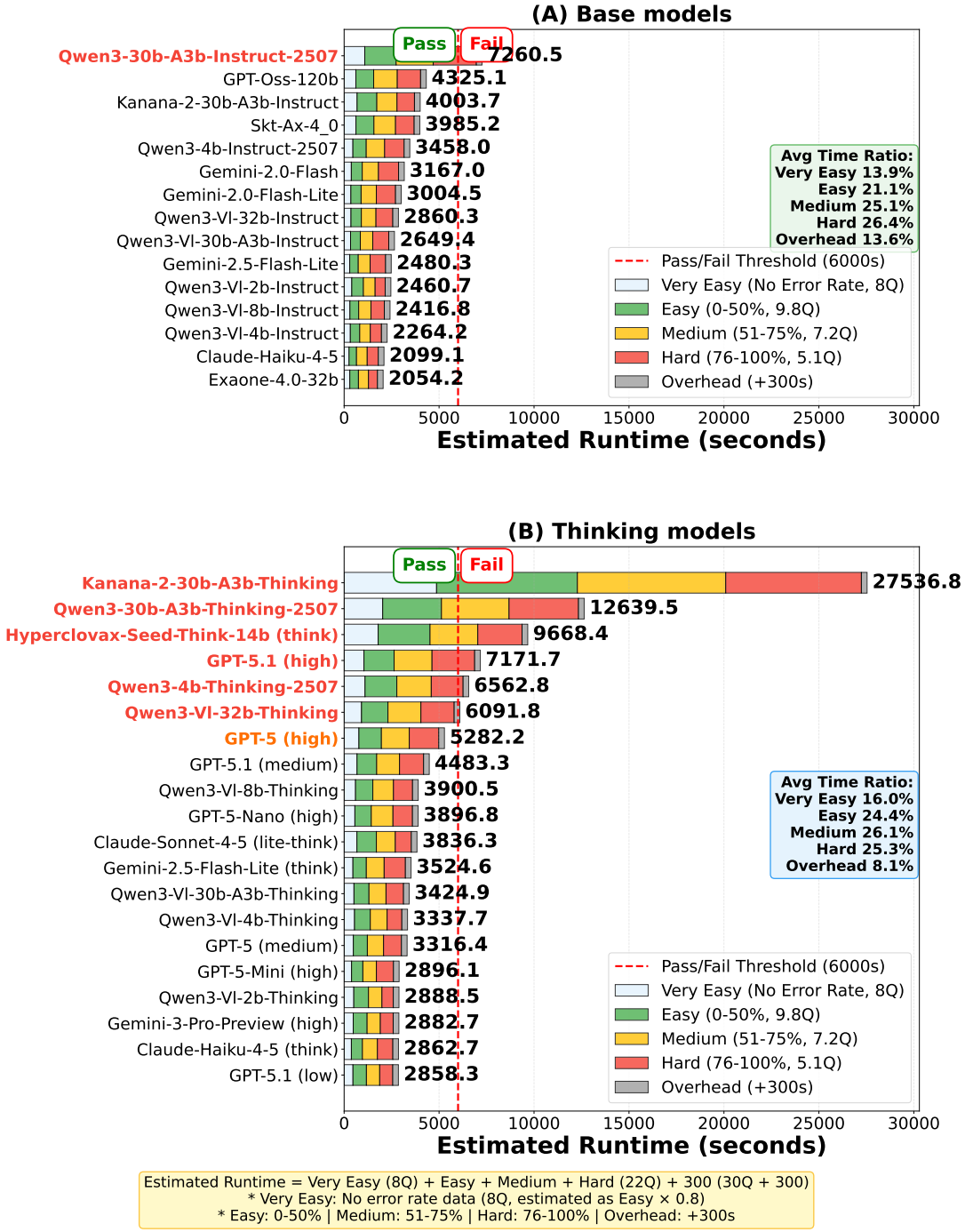



Figure 12: **Estimated wall-clock time to complete a full KCSAT math exam (6000 s limit).** Per-tier average inference latencies weighted by expected question counts (Easy/Medium/Hard plus a "Very Easy" bin for items without published statistics), with a fixed pipeline overhead. (A) Base models; (B) Thinking models. Models left of the dashed line finish within time.

**Prompt**

**# VLM (wo. TTS)**
Please solve the problem given in the image. IMPORTANT: Provide ONLY the final answer. Do NOT show your work, reasoning, thinking process, step-by-step solution, or any explanation. Just give the answer directly.

**# VLM (w. TTS)**
Please solve the problem given in the image.

**# LLM (wo. TTS)**
ocr output: {OCR TEXT}
Please solve the problem given above. IMPORTANT: Provide ONLY the final answer. Do NOT show your work, reasoning, thinking process, step-by-step solution, or any explanation. Just give the answer directly.

**# LLM (w. TTS)**
ocr output: {OCR TEXT}
Please solve the problem given above.

**# Evaluation**
Evaluate whether the predicted answer ('{PRED_TEXT}') matches the correct answer ('{ANSWER_TEXT}'). Respond only with 'Correct' or 'Incorrect'.

Figure 13: **Prompt templates used in experiments.** VLM and LLM settings under wo. TTS and w. TTS, plus the equivalence-judge prompt for answer verification.

**Sample of Structured Metadata for KCSAT-ML**

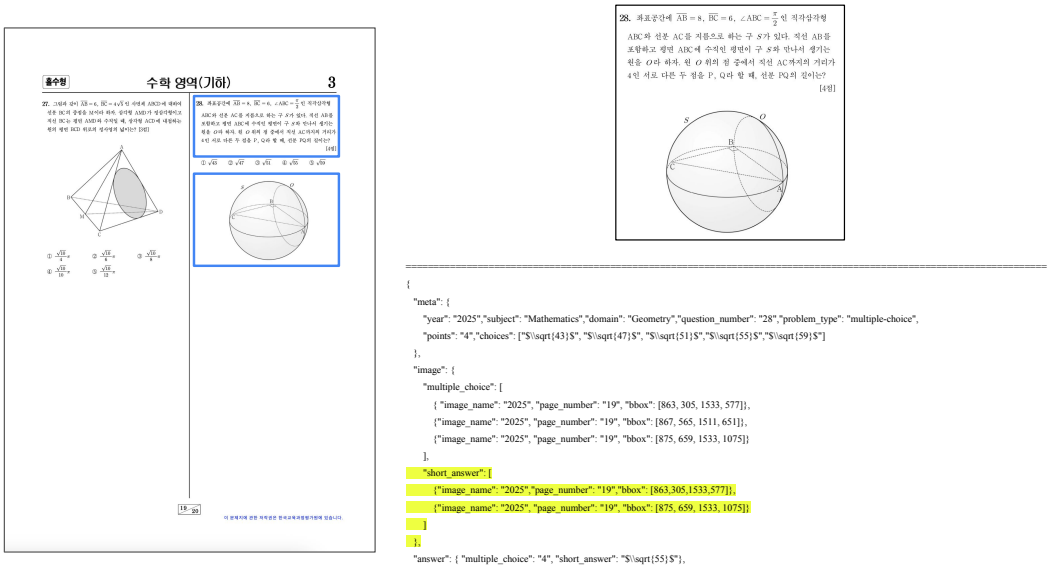



Figure 14: **Sample of structured metadata.** A geometry item in KCSAT-ML showing the original exam page, cropped image regions, and parsed fields (attributes, bounding boxes, answers, human error statistics).