GAP: GENERATION-ALIGNED PAIRED FIDELITY EVALUATION FOR GGUF-QUANTIZED VLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reliably distinguishing small fidelity differences between quantized candidates is challenging during quantization development. We introduce **GAP**, a **G**eneration-**A**ligned **P**aired fidelity evaluation protocol for GGUF-quantized vision-language models. GAP generates image-conditioned responses with the full-precision reference, teacher-forces all candidates on these shared responses, and applies item-level paired tests to their fidelity divergence (e.g., KLD) from the reference to detect small fidelity differences with quantified uncertainty. A pilot-based procedure selects mode-specific token caps to reduce per-item evaluation cost, enabling larger samples and greater statistical power within a fixed budget. Across 126 quantized checkpoints near the Q4 regime, six benchmarks, and 15 model–mode configurations, the median paired-to-unpaired SNR ratio is 3.7 in thinking mode and 2.0 in non-thinking mode for candidate pairs with relative fidelity gaps below 5%; pairing reduces the estimated median sample size per candidate required for 80% power by approximately $15.5\times$ relative to an independent-item unpaired design for such pairs in thinking mode, whereas close non-thinking pairs still need an estimated 11,000 items, beyond our budget. Fidelity rankings can also differ between thinking and non-thinking modes, while responses generated by a different reference model can change verdicts, including statistically significant reversals. These findings support evaluating candidates in their intended inference mode using responses from their corresponding reference model. We release the associated artifacts on Hugging Face Hub: `https://huggingface.co/datasets/Anonymous778862/anony778862cr`.

1 INTRODUCTION

Quantization reduces the memory and computation required to deploy large language models and vision-language models (VLMs) (Xiao et al., 2023; Lin et al., 2024). Under a given deployment budget, a developer can choose among quantization algorithms and calibration procedures depending on the hardware resources. During quantization development, the practical question is whether a new quantized variant is closer to the full-precision model than an existing one.

Llama.cpp (Gerganov et al., 2023) supports a spectrum of weight-only quantization algorithms with similar bits per weight and benefits from an active community of contributors (Han et al., 2023) who make GGUF-quantized models available on Hugging Face, making it a well-suited platform for studying the quantization algorithms under comparable deployment budgets.

In addition to the benchmark scores, distributional fidelity metrics like Kullback–Leibler divergence directly measure the closeness between the quantized variants and the full-precision model, providing a more nuanced signal to differentiate the quantized candidates (Dutta et al., 2024; Nikolić et al., 2026). However, how to evaluate vision-language models using distributional fidelity metrics remains unexplored.

We propose **GAP**, a **G**eneration-**A**ligned **P**aired fidelity evaluation for GGUF-quantized VLMs. GAP uses the full-precision model as a reference to generate on-policy responses for each image–question pair, teacher-forces every quantized variant on these same responses, and compares two variants using an item-level paired test to enhance the discriminative power. Finally, it reports either a statistically significant verdict or an explicit unresolved outcome on the given reference dataset

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.

Across 118 Q4-regime and eight Q3-regime GGUF checkpoints, six benchmarks, and 15 model–mode configurations, pairing reduces the estimated median number of items per candidate needed for 80% power for close pairs by roughly $15.5\times$ in thinking mode relative to an independent-item unpaired design. We further find that candidate ordering can change across inference modes and when the reference responses come from a different model, so close comparisons should be made in the intended mode and on responses from the candidates' own reference.

Our contributions are:

- **GAP**, a paired fidelity-evaluation protocol for determining which of two close quantized candidates is closer to the reference, while explicitly retaining unresolved comparisons.
- A pilot-based procedure for selecting mode-specific token caps that bound per-item scoring cost while preserving statistical discriminability.
- An empirical study showing the statistical efficiency of pairing, how inference mode and reference-response source affect close fidelity comparisons, and that a fidelity improvement in one inference mode need not carry over to another for a pair of quantized candidates, together with released evaluation artifacts.

2 Related Work

Quantized model evaluation. LLM quantization work routinely reports perplexity and KL divergence alongside task accuracy (Dettmers et al., 2022; Frantar et al., 2023; Xiao et al., 2023; Lin et al., 2024; Tseng et al., 2026), whereas quantized VLMs are evaluated through downstream task scores (Wang et al., 2024a; Yu et al., 2025; Li et al., 2025; Xue et al., 2025; Deng & Huang, 2026). Task scores are computionally expensive when each candidate requires free-form generation and scoring. They are also coarse: models with near-identical accuracy can differ markedly in perplexity (Yao et al., 2022; Frantar et al., 2023; Jin et al., 2024). Perplexity, in turn, is cheap to compute but measures fit to a text corpus rather than agreement with the reference model.

Reference-model fidelity. A growing body of work compares quantized models directly with their references. At the distribution level, prior work uses Kullback-Leibler divergence (KLD) (Dutta et al., 2024; Nikolić et al., 2026; Lotfi et al., 2026), bounded divergences such as total variation distance and Jensen–Shannon divergence (Qamar et al., 2026), or the expected acceptance rate‘(Helcig et al., 2026). At the decision level, prior work measures correctness flips (Dutta et al., 2024) or joint correctness via the Correctness Agreement of Rababah et al. (2026). These studies show that fidelity metrics resolve differences hidden by measures such as accuracy and perplexity, but distributional fidelity need not predict downstream performance: KLD tracks how many answers change but not in which direction, and loses its correlation with task scores in the near-lossless regime (Nikolić et al., 2026). Fidelity is therefore generally positioned as a complementary diagnostic alongside task scores or as the criterion for lossless quantization (Helcig et al., 2026).

Choice of context for evaluation. Fidelity can be measured on a fixed corpus or on model-generated continuations. Deiseroth et al. (2024) greedily decode the reference on WikiText-2 prefixes and locate the first top-1 disagreement; Helcig et al. (2026) teacher-force their distribution-lossless quantized models on greedy reference responses to 15 diverse prompts; Nikolić et al. (2026) measure response-only KLD over on-policy reference continuations of instruction-tuning prompts. In contrast, Lotfi et al. (2026) teacher-force the reference on each quantized model's own reasoning traces. This setup helps diagnose model-specific failures but produces different continuations across candidates. Therefore, it does not compare candidates under identical prefixes.

Reliability of evaluation. Uncertainty arising from data sampling and stochastic decoding makes it challenging to obtain reliable estimates of LLM performance. Prior work has explored statistical techniques to account for this uncertainty and improve the reliability of evaluation results (Miller, 2024; Madaan et al., 2024; Zhang et al., 2026; Kübler et al., 2026; Kotawala, 2026). In our work, we use a small pilot set to select evaluation token caps for image-conditioned datasets. By reducing the computational cost per item, this approach makes it feasible to evaluate more samples within a given budget, thereby increasing statistical power to distinguish between two quantized candidates with similar fidelity.

3 METHODOLOGY

The GAP evaluation protocol proceeds in four stages: (i) generating shared reference responses, (ii) collecting fidelity divergences under teacher forcing, (iii) running a paired statistical test on these metrics over image-conditioned responses, and (iv) issuing a verdict. Sections 3.1–3.4 define the evaluation target, the fidelity divergence measuring the closeness of a quantized candidate to the reference, the paired inference procedure and verdict rules, and a signal-to-noise (SNR) analysis of paired fidelity differences between two candidates.

3.1 SHARED REFERENCE RESPONSES AND THE EVALUATION OBJECTIVE

Let f denote a reference model configuration: an unquantized checkpoint θ_{ref} together with its tokenizer, image processor, chat template, and inference mode (thinking or non-thinking, and reasoning strength). Its quantized variants $\mathcal{Q}_f = \{\theta_1, \ldots, \theta_K\}$ share f's tokenizer, processor, chat template and inference mode, which allows their next-token distributions to be aligned with the reference's, position by position. $\mathcal{Q}_f$ represents the quantized variants a developer produces while tuning a quantization algorithm under a range of bits-per-weight constraints.

Let x denote an image-conditioned question. Questions are sampled from a benchmark dataset, which we treat as a sample from the population question distribution $\mathcal{X}$, and x_i denotes the i-th sampled question. For each x_i the reference model generates a single response

$$y_i \sim G_f(\cdot \mid x_i; L_{\text{G}}), \qquad y_i = (y_{i,1}, \ldots, y_{i,L_i}), \tag{1}$$

where G_f denotes the reference model's decoding procedure, L_{G} is the generation length budget, and $L_i \leq L_{\text{G}}$ is the length of the resulting response. We set G_f to the decoding configuration recommended by the model developer, so that the reference responses follow the distribution the model produces in deployment. Responses containing repetition loops (Holtzman et al., 2020) are discarded, so G_f is this distribution conditioned on the absence of such loops within L_{G} tokens. The same y_i is reused for every candidate in $\mathcal{Q}_f$, and only its first

$$m_i(T) = \min\{T, L_i\} \tag{2}$$

tokens are evaluated, where T is the *evaluation token cap*. An end-of-sequence token emitted within this prefix is included, and shorter responses are not padded.

Evaluation objective. Given any pair of candidates $\theta_a, \theta_b \in \mathcal{Q}_f$, our goal is to determine which is closer to the reference model θ_{ref} on the shared prefixes $(x_i,\ y_{i,\leq m_i(T)})$, as measured by the fidelity divergence defined in Section 3.2.

3.2 TEACHER-FORCED FIDELITY EVALUATION

Teacher forcing. For any checkpoint θ, quantized or not, we write $p_\theta(\cdot \mid x, y_{<t})$ for its next-token distribution given a chat-templated question x, which includes the input image(s) and the response prefix $y_{<t} = (y_1, \ldots, y_{t-1})$. We evaluate candidates under *teacher forcing*: rather than letting each candidate decode its own response, we feed it the reference response y_i from Eq. (1) and read off its next-token distribution at every position. At position t of item i this yields $P_{i,t} = p_{\theta_{\text{ref}}}(\cdot \mid x_i, y_{i,<t})$ and $Q^\theta_{i,t} = p_\theta(\cdot \mid x_i, y_{i,<t})$, both taken as the unit-temperature softmax over the full shared vocabulary $\mathcal{V}$. Because every model is conditioned on the same prefix, the two distributions are directly comparable at each position.

Fidelity divergence. A per-position *fidelity divergence* $d(P, Q) \geq 0$ measures how far a candidate distribution Q is from the reference distribution P, with $d(P, Q) = 0$ when $Q = P$ and smaller values indicating greater fidelity to the reference. The construction below applies to any such d. In this paper, we use the Kullback–Leibler divergence (KLD), measured in nats, because it directly quantifies discrepancies between the candidate and reference distributions, consistent with the goal of reproducing the reference model's behavior through quantization.

$$d(P, Q) = D_{\text{KL}}(P \,\|\, Q) = \sum_{v \in \mathcal{V}} P(v) \log \frac{P(v)}{Q(v)}. \tag{3}$$

The item-level fidelity divergence and its corpus-level mean over N evaluated items are

$$D_i(\theta; T) = \frac{1}{m_i(T)} \sum_{t=1}^{m_i(T)} d(P_{i,t}, Q^{\theta}_{i,t}), \qquad \overline{D}(\theta; T) = \frac{1}{N} \sum_{i=1}^{N} D_i(\theta; T). \tag{4}$$

Every item has equal weight regardless of its length. An item is *retained* at cap T when all of its per-position values $d(P_{i,t}, Q^{\theta}_{i,t})$ are finite for all candidates, which ensures that all candidates are evaluated on the same set of N retained items.

3.3 Item-level paired analysis

Paired differences. Following Miller (2024), we analyze item-level paired differences. For a candidate pair $(\theta_a, \theta_b) \in \mathcal{Q}_f^2$ evaluated with the same fidelity divergence, inference mode, dataset, and cap T, the paired difference on item i is

$$\Delta_i(T) = D_i(\theta_a; T) - D_i(\theta_b; T). \tag{5}$$

Because a smaller divergence means being closer to the reference, $\Delta_i > 0$ indicates that θ_b is closer to the reference on item i, and $\Delta_i < 0$ that θ_a is; the labels a and b are arbitrary. Both terms of Δ_i are computed on the same question x_i, the same reference response y_i, and the same prefix length $m_i(T)$, which is what makes the difference a paired observation.

Estimand. Write $D(\theta; x, y, T)$ for the item-level fidelity divergences of Eq. (4) evaluated on an arbitrary question x and response y, so that $D_i(\theta; T) = D(\theta; x_i, y_i, T)$. The quantity we wish to estimate is the expected gap

$$\mu_{ab}(T) = \mathbb{E}_{x \sim \mathcal{X}} \, \mathbb{E}_{y \sim G_f(\cdot|x; L_G)} \big[D(\theta_a; x, y, T) - D(\theta_b; x, y, T) \big], \tag{6}$$

where the outer expectation is over questions and the inner one over the reference model's own responses. The randomness in Δ_i thus has two sources, the sampled question and the sampled reference response. Since we draw a single response per question, the paired analysis takes both into account.

Estimator and confidence interval. Treating the N retained items as i.i.d. draws from the joint distribution in Eq. (6), the point estimate, sample variance, and standard error are

$$\widehat{\mu}_{ab} = \frac{1}{N} \sum_{i=1}^{N} \Delta_i, \qquad s_{ab}^2 = \frac{1}{N-1} \sum_{i=1}^{N} (\Delta_i - \widehat{\mu}_{ab})^2, \qquad \widehat{\mathrm{SE}}_{ab} = \frac{s_{ab}}{\sqrt{N}}. \tag{7}$$

Writing D_a and D_b for $D_i(\theta_a; T)$ and $D_i(\theta_b; T)$, we have $\mathrm{Var}(D_a - D_b) = \mathrm{Var}(D_a) + \mathrm{Var}(D_b) - 2\,\mathrm{Cov}(D_a, D_b)$, so pairing reduces variance whenever the two candidates' item-level fidelity divergences are positively correlated across items. The covariance is estimated from the dataset, and Section 5.4 compares the required sample sizes of paired and independent-item unpaired designs using moments estimated from the observed paired data.

The paired t-test assumes the Δ_i are i.i.d. and compares $t = \widehat{\mu}_{ab} / \widehat{\mathrm{SE}}_{ab}$ against a t distribution with $N - 1$ degrees of freedom; the resulting confidence interval has exact coverage when the differences are normally distributed and approximate coverage otherwise. We report the two-sided 95% confidence interval ($\alpha = 0.05$),

$$\widehat{\mu}_{ab} \;\pm\; t_{1-\alpha/2,\, N-1} \, \widehat{\mathrm{SE}}_{ab}. \tag{8}$$

Verdict. If the interval in Eq. (8) lies entirely above zero, the comparison is *resolved in favor of* θ_b; if it lies entirely below zero, *in favor of* θ_a. Otherwise it is *unresolved*: the evidence is insufficient to determine which candidate is closer, which is not a claim of equivalence.

Multiple comparisons. Verdicts use the nominal per-pair test above unless otherwise stated. Sections 5.1 and 5.3 and their corresponding appendices use Holm-adjusted p-values (Holm, 1979), with the comparison families specified there.

Relative gap. Corpus-level fidelity divergence $\overline{D}(\theta; T)$ differs by more than an order of magnitude across models and modes, so absolute gaps are not comparable across pairs. We therefore also report the relative gap

$$\rho_{ab}(T) = \frac{\mu_{ab}(T)}{\frac{1}{2}\big(\mu_a(T) + \mu_b(T)\big)}, \qquad \mu_\theta(T) = \mathbb{E}_{x\sim\mathcal{X}}\,\mathbb{E}_{y\sim G_f(\cdot|x;L_G)}\big[D(\theta; x, y, T)\big], \tag{9}$$

estimated by $\widehat{\rho}_{ab} = \widehat{\mu}_{ab}\big/\frac{1}{2}\big(\overline{D}(\theta_a; T) + \overline{D}(\theta_b; T)\big)$. A value $\widehat{\rho}_{ab} = 0.05$ reads "θ_b is closer to the reference by 5 % of the pair's mean fidelity divergence".

Pair categories. Both the relative gap and the verdict can change with the cap T, so we categorize pairs using a set of caps rather than a single one, on a stated item set S; write $\widehat{\rho}_{ab}(T; S)$ for the relative gap estimated on S. Let $\mathcal{T} = \{128, 256, 512\}$ in non-thinking mode and $\mathcal{T} = \{1024, 2048\}$ in thinking mode. A pair is *close* on S if $|\widehat{\rho}_{ab}(T; S)| < 5\%$ for at least one $T \in \mathcal{T}$; close pairs are further split into *tight* pairs, with $|\widehat{\rho}_{ab}(T; S)| < 2.5\%$ for every $T \in \mathcal{T}$, and *narrow* pairs, the remainder. A pair that is not close is *wide* if $|\widehat{\rho}_{ab}(T; S)| > 10\%$ for every $T \in \mathcal{T}$, and *moderate* otherwise.

3.4 Paired SNR and sample size

The interval in Eq. (8) narrows with $s_{ab}/\sqrt{N}$, so whether a pair is resolved depends on the gap relative to the spread of the item-level differences. We summarize this with the paired signal-to-noise ratio and its plug-in estimate,

$$\mathrm{SNR}_{ab}(T) = \frac{\mu_{ab}(T)}{\sigma_{ab}(T)}, \qquad \widehat{\mathrm{SNR}}_{ab}(T) = \frac{\widehat{\mu}_{ab}(T)}{s_{ab}(T)}, \tag{10}$$

where $\sigma_{ab}(T)$ is the standard deviation of $\Delta_i(T)$. Since $t = \sqrt{N}\,\widehat{\mathrm{SNR}}_{ab}$, a pair with larger $|\mathrm{SNR}_{ab}|$ is resolved with fewer items; as with $\widehat{\rho}_{ab}$, comparisons across pairs use $|\widehat{\mathrm{SNR}}_{ab}|$. The same ratio drives the token-cap selection of Section 5.1.

How many items does a paired t-test need to resolve a gap of a given size? Under independent normal differences, the paired-t statistic on n items follows a noncentral t distribution with $n - 1$ degrees of freedom and noncentrality $\sqrt{n}\,\mathrm{SNR}_{ab}$, from which the two-sided power $\pi_t(n, \mathrm{SNR}_{ab}; \alpha)$ follows (Appendix L). The plug-in number of items for the conventional 80 % power (Julious, 2004) is

$$\widehat{N_{80}} = \min\{n \geq 2 : \pi_t(n, |\widehat{\mathrm{SNR}}_{ab}|; \alpha) \geq 0.8\}, \tag{11}$$

which grows as $1/|\widehat{\mathrm{SNR}}_{ab}|^2$ in large samples: halving the SNR quadruples the items needed. Inverting the same relation at a fixed budget n gives the smallest relative gap detectable with 80 % power (Appendix L, Table 18).

4 Experimental Setup

Models for evaluation. We evaluate eight models: Qwen3.5-4B, Qwen3.6-35B-A3B, Gemma-4-E4B-it, Gemma-4-31B-it, Kimi-VL-A3B, InternVL3.5-30B-A3B, GLM-4.6V-Flash and Muse-Glimmer-30B. Three of these models are mixture-of-experts [1]. Most of them support both thinking and non-thinking modes. Muse-Glimmer-30B is thinking-only. Kimi-VL-A3B provides separate checkpoints for each mode, giving nine reference checkpoints and 126 quantized checkpoints: 118 in the Q4 regime and eight Q3 variants of comparable size. Since many model developers release Q4 checkpoints[2], we focus on this regime to study how subtle fidelity differences among quantized candidates can be detected (Appendix M lists every revision).

Reference models. Each model–mode configuration has a single shared reference checkpoint. For most models it is the BF16 GGUF released by bartowski on Hugging Face; as no bartowski repository exists for the Kimi-VL-A3B series, we convert Kimi-VL-A3B to BF16 GGUF ourselves with llama.cpp.

[1] Qwen3.6-35B-A3B, Kimi-VL-A3B, and InternVL3.5-30B-A3B are mixture-of-experts

[2] Hugging Face Hub: Qwen/Qwen3-VL-4B-Thinking-GGUF; google/gemma-4-E4B-it-qat-q4_0-gguf

Dataset sampled from diverse benchmarks. Six benchmarks – MMMU-Pro (vision), MATH-Vision, MMStar, OCRBench, OCRBench v2 and BLINK – cover visual perception and reasoning, all chosen for questions that cannot be answered without the image (Appendix H). From each benchmark we randomly sample 500 image-conditioned questions; the first 100 form the pilot set and the remaining 400 form the main set whose size is set by our compute budget. We conduct some preliminary experiments on the pilot set and use the main set for discriminative-power-critical experiments or as a held-out validation set. A cleaning pipeline removes items with shared images, so that no image appears in more than one benchmark or in both the pilot and main sets, leaving 2,971 of the original 3,000 items (Appendix I). Practitioners can substitute other benchmarks and sample sizes suited to their application.

Reference trajectory generation. Each reference checkpoint generates responses to questions in both the pilot and main sets. For the pilot set, we use generation budgets of 16,384 tokens for non-thinking mode and 32,768 tokens for thinking mode. For the main set, we use budgets of 1,024 and 4,096 tokens, respectively, selected based on preliminary experiments on the pilot set. The decoding parameters are described in Appendix J.4. We remove responses with repetition loops (Holtzman et al., 2020) using a repetition detector followed by LLM-based verification (Fable 5.1 or GPT-6-Astra). Retention details for the reference datasets are reported in Table 11 in Appendix I.

Fidelity data collection. Pilot responses are evaluated up to a long cap T_{limit} of 2,048 tokens in non-thinking mode and 8,192 in thinking mode, against which Section 5.1 compares the SNR and verdicts of shorter caps. Main-set responses are evaluated up to 1,024 and 2,048 tokens, respectively, which covers every cap used in the analysis. Candidates are compared in all pairs within each model–mode configuration, giving 635 pairs in non-thinking mode and 746 in thinking mode.

Hardware and inference framework. Both reference generation and KLD collection use llama.cpp. Reference responses are generated on RunPod NVIDIA RTX PRO 6000 instances; KLD records are collected on AWS G7e instances with a fixed AMI and llama.cpp runtime (Appendix K).

5 RESULTS

5.1 HOW TO DECIDE THE TOKEN CAP FOR EVALUATION

We analyze the evaluation token cap on the pilot set from two perspectives: the consistency of paired test results between token caps T_i and T_{limit}, and the mean SNR recovery. This experiment is conducted for each benchmark on the pilot set separately, as is the pair categorization. Verdicts are Holm-corrected within each model–mode–benchmark combination at each T_i Section 3.3. From Figure 1, in general, the proportion of unresolved pairs in thinking mode is lower than that in non-thinking mode at the same T_i. Close and non-close pairs behave differently. For close pairs, the mean SNR recovers faster than for non-close pairs, but the unresolved rate does not decrease stably along with increasing evaluation tokens. In addition, for close pairs at some initial values of T_i, the mean SNR recovery is higher than at $2 \times T_i$, but the proportion of the unresolved pairs is also higher; we consider these results still *unstable*. These results motivate us to use more evaluation tokens on close pairs than non-close pairs. The SNR recovery also differs across benchmarks. To facilitate cross-benchmark comparisons, we use the same evaluation token caps across all benchmarks. We set the evaluation token cap to 256/1,024 in non-thinking/thinking mode for non-close pairs and 512/2,048 for close pairs in subsequent experiments. Appendix E supports these choices: Figure 7 compares each cap with the next one instead of T_{limit}, and Figures 8–11 report the share of pairs whose mean sign at T_i is reversed relative to their verdict at T_{limit}.

5.2 THE SOURCE OF REFERENCE DATASET MATTERS

To examine the effect of the reference dataset source on evaluation using teacher forcing, we conduct a preliminary analysis on the pilot set using Qwen3.5-4B, Qwen3.6-35B-A3B, Gemma-4-E4B-it, and Gemma-4-31B-it as the tested models. All four models are evaluated using teacher forcing on reference datasets generated by Qwen3.5-4B, Qwen3.6-35B-A3B, Gemma-4-E4B-it, Gemma-4-31B-it, and GLM-4.6V-Flash. This analysis covers three scenarios: reference datasets generated by the corresponding model, another model from the same family, and models from other families. For

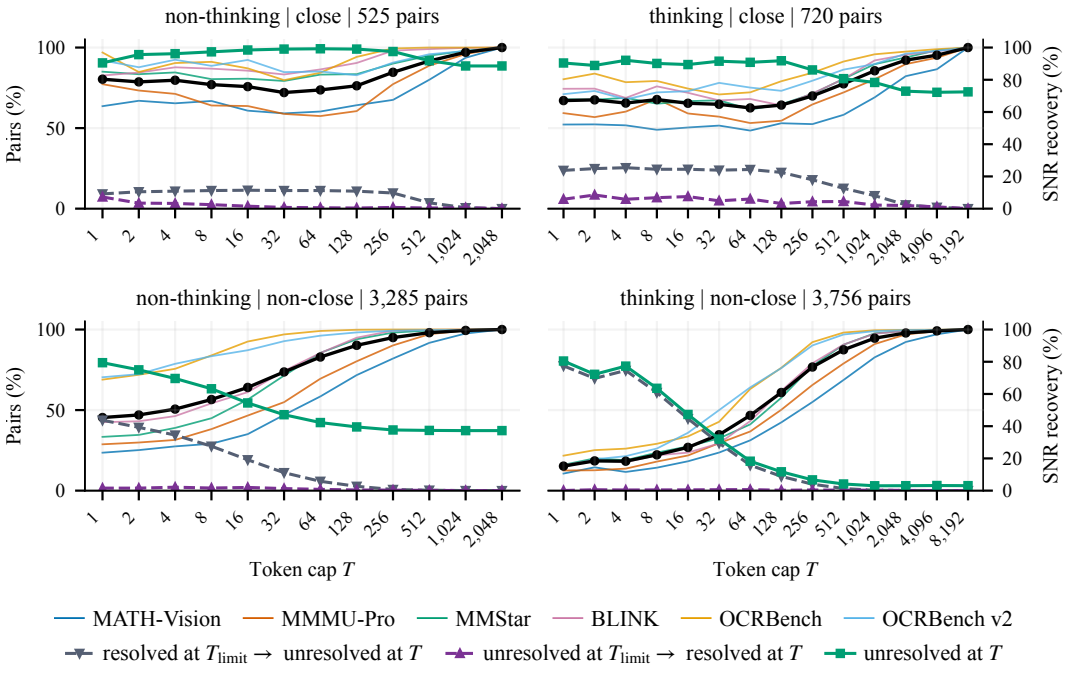

Figure 1: SNR recovery ($|\widehat{\text{SNR}}_{ab}(T_i)|/|\widehat{\text{SNR}}_{ab}(T_{\text{limit}})|$) of evaluation token cap T and verdicts on pilot across models and benchmarks. The thick black line denotes SNR recovery across models and benchmarks. The thin line is per-benchmark SNR recovery across models.

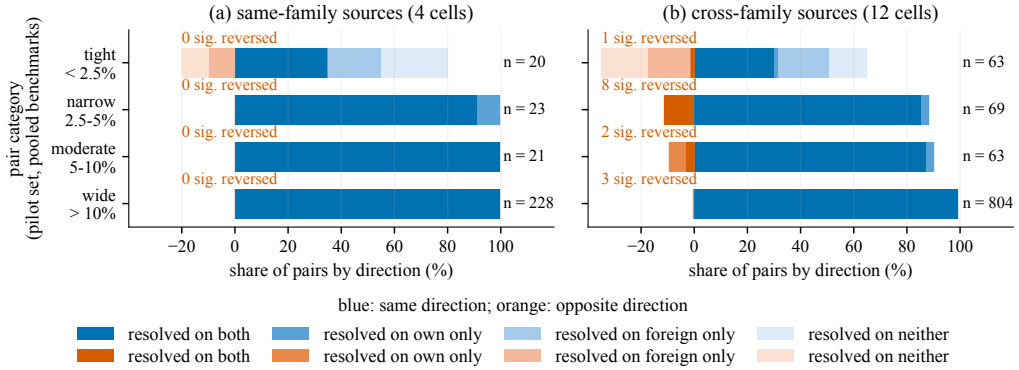

Figure 2: Verdict agreement between a model's own reference dataset and a foreign one; n counts candidate pairs pooled across cells; the pair categories are decided on the pooled pilot set. We only use the reference items shared in corresponding reference dataset and the foreign dataset.

each tested model, the latter two sources are referred to as *foreign* in Figure 2; we call each tested-model/foreign-source combination a *cell*. For Qwen3.5-4B, Qwen3.6-35B-A3B is a same-family but foreign source, and vice versa. The same applies to the two Gemma-4 models. Due to the low discriminative power of the non-thinking pilot set, we only test on thinking mode here.

Figure 2 shows that using the reference dataset generated by a cross-family model can lead to reversed verdicts, particularly for close pairs. Nine out of 132 pairs show statistically significant reversals, approximately 7%; among non-close pairs, the corresponding proportion is 5/867, less than 1%. Even with the same-family reference datasets, verdicts can also differ, although these reversals are not statistically significant under both reference sources on the pilot set. These results suggest that close quantized pairs should be evaluated on reference datasets generated by their corresponding reference models. Appendix F gives the results with pairs categorized separately for each benchmark.

5.3 QUANTIZATION ON DUAL-MODE MODELS

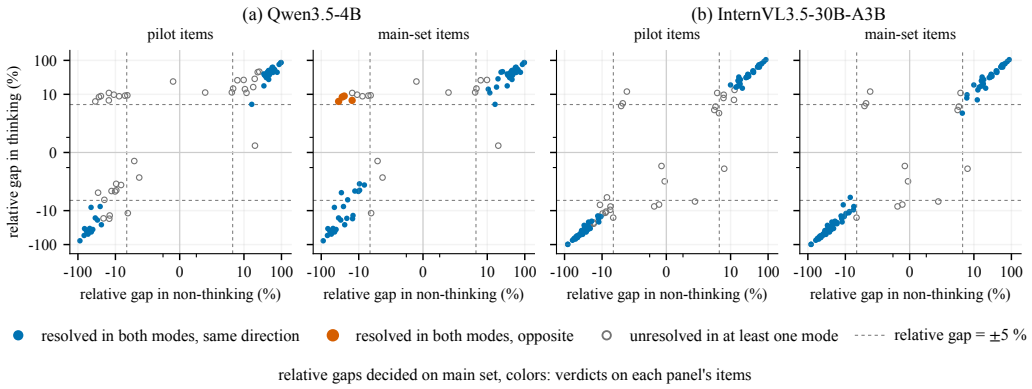

Figure 3: Quantization on dual-mode models, Qwen3.5-4B and InternVL3.5-30B-A3B.

In this section, we discuss whether a quantized variant closer to its reference in one inference mode will also be closer in another mode. To compare results on the pilot and main sets, we use the main set to determine the relative gap so that the point representing each pair has the same position in both comparisons. We use evaluation token caps of 512 and 2,048 for non-thinking and thinking modes, respectively, to get more precise results. The Holm correction is also applied in this experiment across all pairs within each subfigure. Figure 3 shows that being closer in one mode does not always imply being closer in the other. On the main set, five pairs are resolved in opposite directions across modes: four for Qwen3.5-4B and one for Qwen3.6-35B-A3B (Figure 15); none of them is resolved in both modes on the pilot set (Figure 14). To increase discriminative power, we pool the pilot and main sets into the pilot-plus-main set, which shares the same pairs, and identify one additional pair with opposite resolved directions for InternVL3.5-30B-A3B (Figure 13). All six pairs have a relative gap within 10% in at least one mode, and Tables 8 and 9 show that the pilot set cannot detect them, owing to its low discriminative power in non-thinking mode.

5.4 PAIRED VS. UNPAIRED: SNR AND POWER ANALYSIS

Table 1: Median paired-to-unpaired SNR ratio g_{ab} and estimated median $\widehat{N}_{80}$ for paired and unpaired analyses

			Pooled benchmarks		
Mode	Category	Pairs	SNR ratio	$\widehat{N}_{80}$ paired	$\widehat{N}_{80}$ independent
Non-thinking	close	81	2.01	10949 [5155, 13117]	censored [censored, censored]
Non-thinking	non-close	554	1.41	149 [120, 162]	318 [273, 352]
Thinking	close	129	3.68	353 [301, 451]	5471 [4869, 6175]
Thinking	non-close	617	2.02	7 [6, 7]	18 [16, 19]

We use main-set items and pair categories to estimate $\widehat{N}_{80}$ from the observed paired moments. The comparison uses paired noncentral-t power and an independent-item Welch–Satterthwaite approximation (Welch, 1947; Satterthwaite, 1946), with n observations per candidate and $2n$ candidate scores in either design. Within each mode–category group, we report equally weighted lower medians and nominal 95% intervals from 5,000 joint item bootstrap resamples (Appendix L). Table 1 shows larger paired-to-unpaired SNR ratios g_{ab} for thinking and close pairs (defined in Appendix B). Median paired $\widehat{N}_{80}$ estimates range from 7 to approximately 11,000. For close pairs in thinking mode, the median unpaired $\widehat{N}_{80}$ is $15.5\times$ the paired median. For close pairs in non-thinking mode, the median unpaired $\widehat{N}_{80}$ is right-censored under pair-specific limits $L_{ab} = 10N$, where N is the number of retained main-set items for pair (θ_a, θ_b).

Table 2: Median $|\widehat{\mathrm{SNR}}_{ab}(T)|$ on main items over 544 candidate pairs shared by six dual-mode models. Responses are naturally completed and fully scored, with $m_i(T) = L_i$ ($L_i \leq 1{,}024$ in non-thinking and $\leq 2{,}048$ in thinking mode, the extent of the main-set records). Ratios are medians of within-pair ratios. The first two rows pool lengths without matching; the last three match integer-length counts within each model and bin (Appendix C).

Response length L_i	Non-thinking	Thinking	Thinking/non-thinking
All (unmatched)	0.216	1.084	3.769
129–1024 (unmatched)	0.784	1.370	1.360
129–256	0.793	1.006	1.251
257–512	0.907	1.436	1.279
513–1024	0.969	1.676	1.291

5.5 The discriminative capability of thinking and non-thinking reference set

Table 1 leaves close non-thinking pairs beyond the main set: their median paired $\widehat{N_{80}}$ is 10,949 against 2,376 available items, whereas the thinking counterpart is 353. At a fixed cap T, Eqs. (9) and (10) give $\widehat{\mathrm{SNR}}_{ab} = \widehat{\rho}_{ab}/R_{ab}$, with R_{ab} defined in Eq. (19) and dependence on T suppressed here, so in large samples $\widehat{N_{80}} \approx 7.85\, R_{ab}^2/\widehat{\rho}_{ab}^2$ (Eq. (18)). With a median R_{ab} of 1.22 in non-thinking and 0.28 in thinking mode (Table 19), the same relative gap costs about $19\times$ more items in non-thinking mode: roughly 4,700 items for a 5 % gap and 19,000 for a 2.5 % gap, in line with the minimum detectable effects of Table 18. At $N \approx 2{,}400$, unresolved close pairs are the expected outcome.

Table 2 shows where the noise comes from. It uses the same 544 pairs, with every response scored over its full length (up to 1,024 tokens in non-thinking and 2,048 in thinking mode). Without any length control, the median within-pair thinking/non-thinking ratio of $|\widehat{\mathrm{SNR}}_{ab}(T)|$ is 3.77. Keeping only responses of 129–1024 tokens in both modes drops the ratio to 1.36, and matching the length distributions exactly within each bin lowers it only slightly further, to 1.25–1.29. The first step does the work: half of the eligible non-thinking responses are shorter than 129 tokens (Appendix C), and dropping them raises the non-thinking median $|\widehat{\mathrm{SNR}}_{ab}(T)|$ from 0.22 to 0.78, whereas thinking, which loses its short responses and those above 1,024 tokens, moves only from 1.08 to 1.37. This is what to expect if short responses are the noisy ones, since $D_i(\theta; T)$ averages over $m_i(T)$ positions and a short response offers few. The residual $1.3\times$ barely changes across bins whose benchmark mixes differ sharply. The mode gap therefore sits in short non-thinking responses rather than across all lengths. Two caveats apply. The $3.77\times$ excludes thinking responses above 2,048 tokens, which the bins suggest have high $|\widehat{\mathrm{SNR}}_{ab}(T)|$, so it is if anything an underestimate. And the residual cannot be attributed to reasoning content, because at any given length the two modes answer different questions. Since verdicts do not transfer across modes (Section 5.3), a non-thinking deployment must budget 5,000–20,000 items for close pairs; benchmarks whose non-thinking responses run long need fewer, but only if their questions represent the deployment.

6 Conclusion and Discussion

GAP detects fidelity differences between two closely matched quantized candidates with a reference-generated shared corpus, a paired test, and a pilot-selected token cap. An improvement in fidelity in thinking mode does not necessarily carry over to non-thinking mode, or vice versa, so candidates must be evaluated in their deployment mode; in non-thinking mode, close pairs need 5,000–20,000 items (Section 5.5), a budget that benchmarks eliciting long responses can reduce. Fidelity is only one aspect of VLM evaluation and does not always reflect downstream performance, which must still be evaluated before deployment. In practice, a minimum improvement threshold or tolerance should be tested with minimum-effect, non-inferiority, or equivalence tests.

AI USE STATEMENT

We used generative AI agents (Fable 5.1, Opus 5.5, and GPT-6-Astra) to assist with the following tasks:

- **Literature review.** The agents helped identify and search related work. The authors checked the recommended papers and supporting materials directly by reading or browsing to verify the information provided by the agents.
- **Experiment selection.** The agents helped the authors discuss which planned experiments would be most interesting to readers. All experiment plans were developed by the authors; the agents assisted only with selecting among these plans.
- **Mathematical modeling and problem formulation.** The agents assisted with the mathematical modeling and problem formulation for the method presented in Section 3. The authors verified the resulting formulations.
- **Statistical methodology.** The agents surveyed methods for estimating N_{80} under different statistical inference settings, recommended approaches for our setting, and provided reference materials. The authors reviewed these materials to assess the recommendations.
- **Code implementation and review.** The agents assisted with implementing and reviewing the experimental code, including inference code for llama.cpp and code for analyzing fidelity metrics collected during the experiments. The authors verified the implementation by tracing the source code and testing the resulting artifacts.
- **Experiment execution.** The agents wrote scripts and ran experiments using parameters specified by the authors.
- **Reference dataset cleaning.** The agents assisted with cleaning the reference dataset by removing reference responses containing repetition loops.
- **Figure and table preparation.** The agents helped create figures and format tables in LaTeX from the experimental artifacts.
- **Translation and manuscript editing.** The agents translated sentences and paragraphs from other languages into English and iteratively edited the manuscript to improve wording, grammar, and readability.
- **Title refinement.** The authors chose the term "generation-aligned" and used the agents to explore words beginning with the letter "P," such as "protocol" and "procedure," for the title.

Among the tasks with required disclosure, the agents assisted with developing the conceptual framework and mathematical formulation (Section 3), providing feedback on the experimental and statistical methodology, implementing methods, cleaning the reference dataset, and translation, as listed above. We have not used generative AI tools to generate synthetic data sets, propose or refine hypotheses, or interpret results; qualitative and thematic data analysis is not applicable to this work. Among the tasks with recommended disclosure, the agents assisted with literature search and summarization, creating and editing software code and figures, formatting tables, editing the manuscript for readability, and proposing title words. We have reviewed all AI-assisted work: the authors verified every formulation, traced and tested all agent-written code, checked the recommended references against the original papers, re-derived the reported numbers from the released artifacts, and reviewed every edited sentence. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All artifacts are released at the anonymous Hugging Face repository linked in the abstract: the reference response sets for every model–mode configuration, the per-item fidelity records, and the collect_meta.json of every scoring run, which records all runtime settings together with the SHA-256 of the scorer binary, its backend libraries, and both GGUF files. The repository also contains the analysis scripts that regenerate every table and figure from these records, and the scorer as a patch against a pinned upstream llama.cpp commit. Appendix J.3 and K list the runtime settings, Appendix J.4 the decoding parameters, and Appendix M the exact revision of every checkpoint used.

ACKNOWLEDGMENTS

We thank the extensive community of llama.cpp developers, the contributors who release GGUF models and their quantized variants on the Hugging Face Hub, and the authors of the open-source datasets MMMU-Pro, MATH-Vision, MMStar, OCRBench, OCRBench v2, and BLINK. Their contributions made this research feasible.

REFERENCES

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, et al. Are we on the right way for evaluating large vision-language models? In *Advances in Neural Information Processing Systems*, volume 37, pp. 27056–27087, 2024.

Björn Deiseroth, Max Meuer, Nikolas Gritsch, Constantin Eichenberg, Patrick Schramowski, Matthias Aßenmacher, and Kristian Kersting. Divergent token metrics: Measuring degradation to prune away LLM components – and optimize quantization. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6764–6783, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.377. URL https://aclanthology.org/2024.naacl-long.377/.

Juncan Deng and Kejie Huang. VLM-PTQ: Efficient post-training quantization for large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24696–24705, June 2026.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in neural information processing systems*, volume 35, pp. 30318–30332, 2022.

Abhinav Dutta, Sanjeev Krishnan, Nipun Kwatra, and Ramachandran Ramjee. Accuracy is not all you need. In *Advances in Neural Information Processing Systems*, volume 37, pp. 124347–124390, 2024.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, Jingqun Tang, Wei Chen, Lianwen Jin, Yuliang Liu, and Xiang Bai. OCRBench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. BLINK: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pp. 148–166, 2024.

Georgi Gerganov et al. llama.cpp: LLM inference in C/C++, 2023. URL https://github.com/ggml-org/llama.cpp.

Daniel Han, Michael Han, et al. Unsloth, 2023. URL https://github.com/unslothai/unsloth.

Michael Helcig, Eldar Kurtic, and Dan Alistarh. Statistically-lossless quantization of large language models, 2026. URL https://arxiv.org/abs/2605.02404.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. A comprehensive evaluation of quantization strategies for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12186–12215, 2024.

Steven A Julious. Sample sizes for clinical trials with Normal data. *Statistics in Medicine*, 23(12): 1921–1986, 2004.

Anany Kotawala. Resolution diagnostics for paired LLM evaluation. In *The ICML 2026 Workshop on Hypothesis Testing*, 2026. URL https://openreview.net/forum?id=1MY39ELzdX.

Jonas M. Kübler, Kailash Budhathoki, Matthäus Kleindessner, Xiong Zhou, Junming Yin, Ashish Khetan, and George Karypis. When LLMs get significantly worse: A statistical approach to detect model degradations. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=cM3gsqEI4K.

Shiyao Li, Yingchun Hu, Xuefei Ning, Xihui Liu, Ke Hong, Xiaotao Jia, Xiuhong Li, Yaqi Yan, Pei Ran, Guohao Dai, et al. MBQ: Modality-balanced quantization for large vision-language models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4167–4177. IEEE, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pp. 87–100, 2024.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: on the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12), December 2024. ISSN 1869-1919. doi: 10.1007/s11432-024-4235-6. URL http://dx.doi.org/10.1007/s11432-024-4235-6.

Sanae Lotfi, Polina Kirichenko, Steven Li, and Zechun Liu. Quantized reasoning models think they need to think longer, but they do not, 2026. URL https://arxiv.org/abs/2606.00206.

Lovish Madaan, Aaditya K. Singh, Rylan Schaeffer, Andrew Poulton, Sanmi Koyejo, Pontus Stenetorp, Sharan Narang, and Dieuwke Hupkes. Quantifying variance in evaluation benchmarks. arXiv preprint arXiv:2406.10229, 2024. URL https://arxiv.org/abs/2406.10229.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations, 2024. URL https://arxiv.org/abs/2411.00640.

Miloš Nikolić, Ali Hadi Zadeh, Enrique Torres Sanchez, and Andreas Moshovos. Displacement is not direction: Evaluating fidelity metrics for quantized LLM deployment, 2026. URL https://arxiv.org/abs/2606.19558.

Shahzeb Qamar, Lorenz Sparrenberg, Christian Bauckhage, Baha Rababah, Carson Leung, Murat Kantarcioglu, Cuneyt Gurcan Akcora, and Rafet Sifa. Accuracy is not enough: A divergence-based approach to evaluate fidelity loss in quantized LLMs, 2026. URL https://arxiv.org/abs/2609.07664.

Baha Rababah, Shahzeb Qamar, Lorenz Sparrenberg, Rafet Sifa, Murat Kantarcioglu, Cuneyt Gurcan Akcora, and Carson K. Leung. The illusion of equivalency: Statistical characterization of quantization effects in LLMs, 2026. URL https://arxiv.org/abs/2607.08734.

F. E. Satterthwaite. An approximate distribution of estimates of variance components. *Biometrics Bulletin*, 2(6):110–114, 1946. ISSN 00994987. URL http://www.jstor.org/stable/3002019.

Albert Tseng, Zhaofeng Sun, and Christopher De Sa. Model-preserving adaptive rounding. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=PKFilPWjMI.

Changyuan Wang, Ziwei Wang, Xiuwei Xu, Yansong Tang, Jie Zhou, and Jiwen Lu. Q-VLM: Post-training quantization for large vision-language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 114553–114573, 2024a.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. Measuring multimodal mathematical reasoning with MATH-Vision dataset. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95095–95169, 2024b.

B. L. Welch. The generalization of 'student's' problem when several different population varlances are involved. *Biometrika*, 34(1-2):28–35, 01 1947.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099. PMLR, 2023.

Yufei Xue, Yushi Huang, Jiawei Shao, Lunjie Zhu, Chi Zhang, Xuelong Li, and Jun Zhang. VLMQ: Token saliency-driven post-training quantization for vision-language models, 2025. URL https://arxiv.org/abs/2508.03351.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27168–27183, 2022.

JiangYong Yu, Sifan Zhou, Dawei Yang, Shuoyu Li, Shuo Wang, Xing Hu, Chen Xu, Zukang Xu, Changyong Shu, and Zhihang Yuan. MQuant: Unleashing the inference potential of multimodal large language models via full static quantization. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pp. 1783–1792, 2025.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9556–9567, June 2024.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. MMMU-Pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15134–15186, Jul 2025.

Wenbo Zhang, Hengrui Cai, and Wenyu Chen. Beyond the singular: Revealing the value of multiple generations in benchmark evaluation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 10033–10043, July 2026. URL https://aclanthology.org/2026.findings-acl.488/.

A Overview of experimental settings

Table 3: Category sources, evaluation items, benchmark scope, and Holm correction for the main experiments. "Pooled" means that items from the six benchmarks are combined within each model and mode; "per benchmark" means separate statistics for each benchmark. Category sources follow the multi-cap definition in Section 3.3.

Experiment	Pair categories: source and use	Evaluation items	Benchmark scope	Holm
Section 5.1: token-cap curves	Pilot, per benchmark; close / non-close	Pilot	Per benchmark; curves also aggregate comparisons	Yes[a]
Section 5.2: reference corpus (thinking)	Pilot, pooled; tight / narrow / moderate / wide	Pilot; shared own/foreign items in each cell	Pooled	No
Section 5.3: mode transfer	Not used; all pairs common to both modes	Pilot; main; pilot+main[c]	Pooled	Yes[b]
Section 5.4: paired vs. unpaired	Main, pooled; close / non-close	Main	Pooled	No[d]
Section 5.5: power examples	Main, pooled; close groups from Section 5.4; all pairs for R_{ab} summaries	Main	Pooled	No[d]
Section 5.5: full-answer SNR	Main, pooled labels retained as metadata; all 544 common pairs	Main; eligible full answers, with / without length matching	Pooled; matching within model	No[e]

Close combines tight and narrow; non-close combines moderate and wide. Thinking and non-thinking scores are analyzed separately. "No" denotes no multiple-comparison adjustment; nominal tests and power calculations use $\alpha = 0.05$.

[a] Holm adjusts all candidate pairs within each model–mode–benchmark family at each cap, before category filtering. It is not applied jointly across caps. Aggregating the plotted curves does not pool the items for inference.

[b] Holm adjusts all candidate pairs in both modes jointly within each model and verdict item set: $2P$ tests for P common pairs. Pilot, main, and pilot+main are separate families.

[c] The pilot and main panels in Section 5.3 use main-set relative gaps for their coordinates, while verdicts use the panel's stated items. The combined analysis uses pilot+main items.

[d] Required sample sizes use nominal per-pair power. The intervals in Section 5.4 are nominal 95% intervals from 5,000 joint item bootstrap resamples. The fixed-gap examples in Section 5.5 use all-pair median R_{ab} values, not close-only medians.

[e] The SNR table uses all common pairs without significance or category filtering. Pairwise tests computed by the analysis code are nominal; no cross-mode significance test is reported for the SNR ratio.

Table 4: Settings of the supplementary analyses. Benchmark scope describes the statistics, rather than the number of panels or the display of benchmark shares. A dash in the Holm column denotes a descriptive summary without a significance decision.

Analysis	Pair categories: source and use	Evaluation items	Benchmark scope	Holm
Appendix B: covariance and SNR ratios	Not used; all pairs at fixed mode-specific caps	Pilot and main, separately	Pooled	—
Appendix C: length comparison	As Section 5.5; all common pairs	Main; same full-answer cohorts as the main text	Pooled SNR; benchmark counts and shares reported	No[e]
Appendix D: pair-gap plot	Pilot, pooled; categories held fixed to choose caps; all pairs	Pilot and main, separately	Pooled	—
Appendix E: adjacent-cap curves	Pilot, per benchmark; close / non-close	Pilot	Per benchmark	Yes[a]
Appendix E: mean-sign reversals	All categories; pilot/main labels match the item set and benchmark scope	Pilot and main, separately	Pooled and per benchmark, separately	Yes[f]
Appendix F: reference corpus (thinking)	Pilot, per benchmark; four categories	Pilot; shared own/foreign items in each cell	Per benchmark	No
Appendix G: mode transfer	Not used; all common pairs	Pilot; main; pilot+main, separately	Pooled	Yes[b]
Appendix L: $\widehat{N}_{80}$	Main, pooled; close / non-close	Main	Pooled	No[d]
Appendix L: MDE and R_{ab}	Main, pooled categories held fixed to choose caps; all pairs	Main; pilot as a separate sensitivity analysis	Pooled	No[d]

Superscripts a–e follow Table 3. In the mode-transfer appendix, each figure uses its own item set for both verdicts and relative-gap coordinates. Descriptive counts and benchmark shares in the length appendix do not involve multiplicity adjustment.

[f] For pooled reversals, Holm adjusts all pairs within each model–mode family at a fixed cap; per-benchmark reversals additionally separate the families by benchmark. The reversal denominator contains only pairs resolved after Holm at T_{limit}. Reversal at a shorter cap is a change in the mean sign and does not require significance at that shorter cap.

B COVARIANCE ANALYSIS

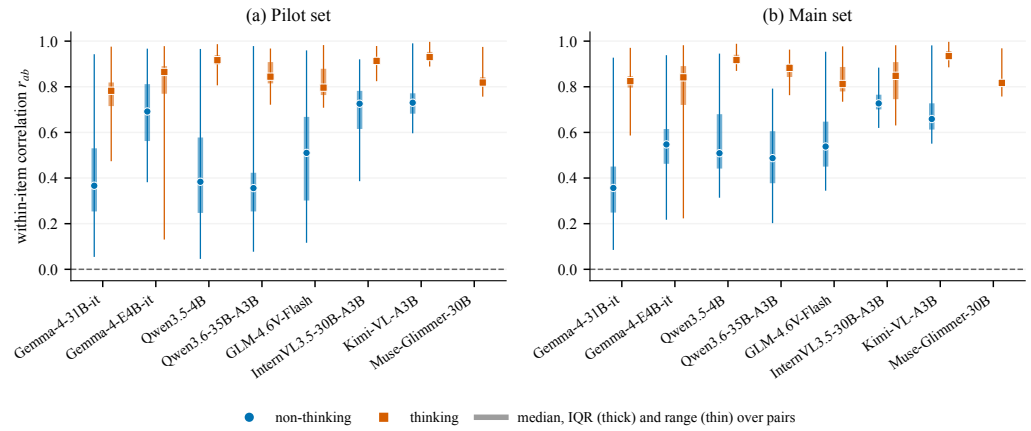

Figure 4: Within-item correlations on pilot and main items.

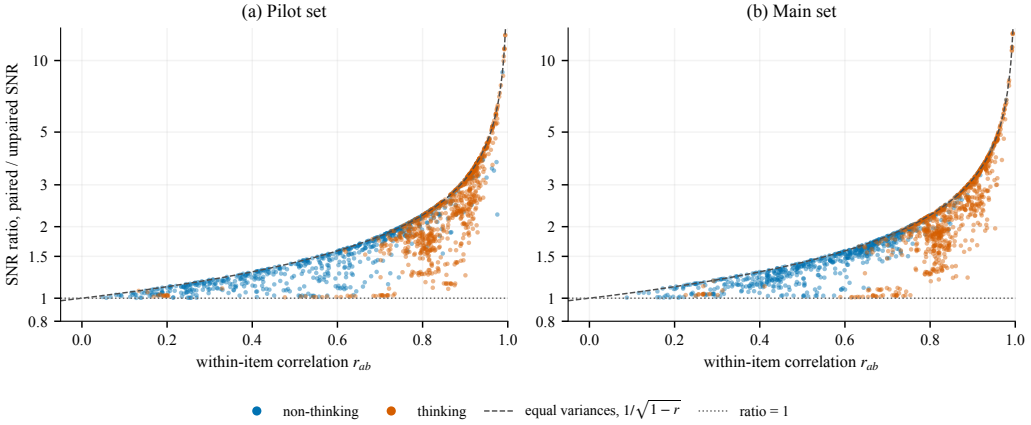

Figure 5: Paired-to-unpaired SNR ratios on pilot and main items.

Figures 4 and 5 pool six benchmarks separately for pilot (left) and main (right) items, using KLD evaluation caps of 512 tokens for non-thinking and 2,048 for thinking. Each item set contains 635 non-thinking and 746 thinking candidate pairs. In Figure 4, r_{ab} is the sample Pearson correlation between $D_i(\theta_a; T)$ and $D_i(\theta_b; T)$ over the same items. Within each model, mode, and item set, markers show the median across pairs, thick bars span the 25th–75th percentiles, and thin bars span the minimum to maximum r_{ab} across all pairs.

Figure 5 shows each pair's paired-to-unpaired SNR ratio. At a fixed cap T, write s_a^2 and s_b^2 for the sample variances of $D_i(\theta_a; T)$ and $D_i(\theta_b; T)$, and s_{ab}^2 for the sample variance of $\Delta_i(T)$. With N observations per candidate, the ratio of independent-item to paired estimated standard errors is

$$g_{ab} = \frac{\widehat{\mathrm{SE}}_{U,ab}}{\widehat{\mathrm{SE}}_{ab}} = \sqrt{\frac{s_a^2 + s_b^2}{s_{ab}^2}} = \frac{1}{\sqrt{1 - r_{ab} h}}, \qquad h = \frac{2 s_a s_b}{s_a^2 + s_b^2} \leq 1.$$

Here $\widehat{\mathrm{SE}}_{ab} = s_{ab}/\sqrt{N}$ as in Eq. (7), and $\widehat{\mathrm{SE}}_{U,ab} = \sqrt{(s_a^2 + s_b^2)/N}$. For a nonzero mean difference and positive scales, g_{ab} also equals $|\widehat{\mathrm{SNR}}_{ab}|/|\widehat{\mathrm{SNR}}_{U,ab}|$, with the independent-item SNR defined in Appendix L. We suppress dependence on the fixed cap T. The dashed curve assumes equal candidate variances ($s_a^2 = s_b^2$, hence $h = 1$), giving $g_{ab} = 1/\sqrt{1 - r_{ab}}$; the curve label abbreviates r_{ab} as r.

For positive r_{ab}, this is an upper bound at each correlation; unequal variances reduce the SNR ratio, explaining points below the curve.

All observed pairs have $r_{ab} > 0$ and $g_{ab} > 1$ on both item sets, so accounting for covariance reduces the estimated standard error in every comparison. Median SNR ratios across pairs are 1.44 and 2.27 on pilot items, and 1.47 and 2.20 on main items, for non-thinking and thinking, respectively. All six checkpoints supporting both modes have higher median correlations in thinking mode on both sets at these evaluation caps. These patterns show that pairing improves precision throughout the evaluated pairs, while variance imbalance limits the benefit of high correlation.

C LENGTH-MATCHED COMPARISON OF THINKING AND NON-THINKING MODE

The comparison in Table 2 uses main items and the 544 candidate pairs shared by both modes of the six dual-mode models. An item is eligible in a model–mode configuration if its reference response terminates naturally within the generation budget and is fully scored by every candidate of that configuration; the main-set records extend to 1,024 tokens in non-thinking and 2,048 in thinking mode, so responses that hit the generation budget or exceed the record length are excluded. Unlike the cap rule of Section 5.1, every score averages the full response, equivalent to $T \geq L_i$ and $m_i(T) = L_i$. The two unmatched cohorts keep all eligible items and those of 129–1024 tokens, respectively, without matching counts or questions across modes.

For the original three bins, each integer length ℓ retains $\min(N_{\mathrm{NT},\ell}, N_{\mathrm{T},\ell})$ items per mode, where $N_{\mathrm{NT},\ell}$ and $N_{\mathrm{T},\ell}$ count eligible responses of length ℓ in non-thinking and thinking mode. Selection ranks the SHA-256 digest of `08|0|model|mode|item_key`, breaking ties by `item_key`; the key contains benchmark, main phase and item ID. This reproduces the original selected cohorts and all three matched SNR rows. Each model–bin has 67–258 items per mode. Each candidate pair uses the same items within a mode; the two modes may use different questions. The SNR magnitude is $|\widehat{\mathrm{SNR}}_{ab}(T)| = |\widehat{\mu}_{ab}(T)|/s_{ab}(T)$, with sample standard deviation computed using $N - 1$ for the N retained items of that pair and mode. Summaries give every candidate pair equal weight and form thinking/non-thinking ratios within pair before taking their median. Tables 5 and 6 list retained counts and benchmark shares.

The pooled counts sum model–item observations and do not deduplicate questions shared across models. Thus item-count pooling differs from the pair weighting used for SNR. Before matching, the eligible cohorts contain 12,275 non-thinking and 9,235 thinking model–item observations. Restricting to 129–1024 retains 6,076 and 6,507, respectively. This removes 6,199 short non-thinking responses and 658 short plus 2,070 long thinking responses. Consequently, the change from the all-length to the restricted baseline does not isolate the contribution of short items alone. Table 7 distinguishes short-item prevalence in all aligned main records from prevalence after natural-completion and full-scoring eligibility.

Exact length matching does not balance benchmark composition. For example, OCRBench v1 comprises 7.8% of the pooled non-thinking 129–256 cohort versus 50.8% of its thinking counterpart. In 513–1024, MATH-Vision accounts for 39.5% versus 2.4%. Questions, content and benchmark composition therefore remain potential explanations for the SNR differences; the residual ratio is not an estimate of a reasoning-content effect. The absolute-value transformation can also bias noisy effect estimates upward near zero, but it does not imply a guaranteed direction of bias for a ratio between modes.

Table 5: Non-thinking item counts and retained benchmark shares (%). Before counts eligible full answers in the specified cohort; after is the retained count. U denotes unmatched; the three shorter ranges are length-matched. All pools model–item observations. MV: MATH-Vision; MP: MMMU-Pro; MS: MMStar; BL: BLINK; O1/O2: OCRBench v1/v2.

Model	Cohort	Before	After	MV	MP	MS	BL	O1	O2
All models	All	12275	12275	10.7	14.8	18.0	19.0	19.2	18.3
All models	129–1024 U	6076	6076	20.9	25.9	19.0	18.6	3.4	12.1
All models	129–256	2229	1167	6.9	14.6	28.4	29.0	7.8	13.2
All models	257–512	2218	1280	22.0	32.8	15.4	15.3	1.9	12.7
All models	513–1024	1629	760	39.5	36.8	8.8	4.6	0.1	10.1
Gemma 31B	All	2210	2210	13.5	16.2	17.4	18.0	17.8	17.1
Gemma 31B	129–1024 U	972	972	30.2	31.6	17.8	8.2	0.4	11.7
Gemma 31B	129–256	292	200	8.5	21.5	34.0	19.0	1.5	15.5
Gemma 31B	257–512	370	253	27.7	38.7	15.0	6.3	0.0	12.3
Gemma 31B	513–1024	310	148	50.0	33.8	6.8	1.4	0.0	8.1
Gemma E4B	All	1902	1902	6.9	13.6	18.9	20.4	20.7	19.4
Gemma E4B	129–1024 U	1018	1018	12.8	24.4	23.6	23.1	2.8	13.4
Gemma E4B	129–256	327	162	1.9	4.3	34.0	45.1	4.9	9.9
Gemma E4B	257–512	349	240	10.0	28.8	23.8	18.3	2.1	17.1
Gemma E4B	513–1024	342	214	28.5	40.2	13.1	5.6	0.5	12.1
GLM Flash	All	2247	2247	15.4	15.3	17.1	17.6	17.5	17.1
GLM Flash	129–1024 U	994	994	33.9	30.7	14.1	4.7	5.9	10.7
GLM Flash	129–256	330	238	10.1	19.7	27.3	9.7	16.4	16.8
GLM Flash	257–512	376	188	33.5	37.8	11.2	4.8	2.1	10.6
GLM Flash	513–1024	288	125	56.0	32.8	4.0	1.6	0.0	5.6
InternVL 30B	All	2302	2302	15.4	16.7	16.8	17.3	17.1	16.6
InternVL 30B	129–1024 U	894	894	36.8	28.1	13.0	8.1	1.0	13.1
InternVL 30B	129–256	298	156	18.6	25.6	21.8	17.9	1.9	14.1
InternVL 30B	257–512	340	258	39.1	33.3	8.9	7.0	0.8	10.9
InternVL 30B	513–1024	256	116	56.0	30.2	6.0	0.0	0.0	7.8
Qwen3.5 4B	All	1782	1782	4.8	12.5	19.4	21.1	22.0	20.3
Qwen3.5 4B	129–1024 U	1075	1075	7.7	20.5	23.2	31.9	5.0	11.7
Qwen3.5 4B	129–256	476	192	2.1	7.8	29.7	41.1	9.4	9.9
Qwen3.5 4B	257–512	390	169	7.7	26.0	20.1	30.2	4.1	11.8
Qwen3.5 4B	513–1024	209	67	25.4	41.8	10.4	10.4	0.0	11.9
Qwen3.6 35B	All	1832	1832	5.2	13.5	18.8	20.7	21.5	20.3
Qwen3.6 35B	129–1024 U	1123	1123	8.5	21.8	21.1	31.7	4.7	12.2
Qwen3.6 35B	129–256	506	219	1.8	8.2	24.2	44.7	9.1	11.9
Qwen3.6 35B	257–512	393	172	5.8	30.2	14.0	33.7	3.5	12.8
Qwen3.6 35B	513–1024	224	90	14.4	44.4	11.1	13.3	0.0	16.7

Table 6: Thinking item counts and retained benchmark shares (%), with the same definitions as Table 5.

Model	Cohort	Before	After	MV	MP	MS	BL	O1	O2
All models	All	9235	9235	4.9	11.0	19.9	19.3	23.9	21.0
All models	129–1024 U	6507	6507	1.8	7.5	21.7	20.7	27.2	21.1
All models	129–256	1664	1167	0.3	0.2	13.5	14.0	50.8	21.3
All models	257–512	2591	1280	0.9	4.0	26.3	20.3	27.4	21.1
All models	513–1024	2252	760	2.4	19.2	22.4	25.4	10.5	20.1
Gemma 31B	All	1732	1732	7.0	12.5	19.7	19.1	21.9	19.7
Gemma 31B	129–1024 U	1317	1317	2.9	8.0	21.0	20.4	27.6	20.1
Gemma 31B	129–256	321	200	0.5	0.0	16.0	21.5	42.5	19.5
Gemma 31B	257–512	571	253	0.8	2.4	24.1	21.3	28.5	22.9
Gemma 31B	513–1024	425	148	4.1	21.6	23.6	22.3	10.8	17.6
Gemma E4B	All	1925	1925	7.5	13.8	18.6	20.2	20.5	19.5
Gemma E4B	129–1024 U	1396	1396	1.8	7.7	22.3	21.8	27.4	19.0
Gemma E4B	129–256	242	162	0.6	0.0	4.3	0.6	69.1	25.3
Gemma E4B	257–512	530	240	0.4	1.2	27.5	21.2	32.5	17.1
Gemma E4B	513–1024	624	214	1.9	15.0	24.3	30.4	10.3	18.2
GLM Flash	All	1813	1813	4.9	13.5	19.9	20.7	21.6	19.4
GLM Flash	129–1024 U	1004	1004	3.0	14.2	25.4	29.4	9.0	19.0
GLM Flash	129–256	372	238	0.0	0.8	30.3	37.8	15.1	16.0
GLM Flash	257–512	313	188	3.7	16.5	25.0	26.1	6.4	22.3
GLM Flash	513–1024	319	125	2.4	34.4	14.4	25.6	5.6	17.6
InternVL 30B	All	1570	1570	3.4	9.4	20.3	22.4	23.8	20.7
InternVL 30B	129–1024 U	1231	1231	1.0	7.6	20.3	21.9	28.8	20.6
InternVL 30B	129–256	242	156	0.0	0.0	16.0	5.8	55.8	22.4
InternVL 30B	257–512	612	258	0.4	3.5	27.1	25.6	24.0	19.4
InternVL 30B	513–1024	377	116	2.6	23.3	15.5	25.0	19.8	13.8
Qwen3.5 4B	All	990	990	1.7	5.5	22.2	15.3	31.5	23.8
Qwen3.5 4B	129–1024 U	718	718	0.7	2.6	21.4	12.8	38.3	24.1
Qwen3.5 4B	129–256	230	192	0.0	0.0	5.2	5.2	69.3	20.3
Qwen3.5 4B	257–512	274	169	0.0	0.6	26.0	11.8	35.5	26.0
Qwen3.5 4B	513–1024	214	67	1.5	9.0	35.8	14.9	11.9	26.9
Qwen3.6 35B	All	1205	1205	2.5	6.8	19.8	15.1	29.7	26.1
Qwen3.6 35B	129–1024 U	841	841	0.7	2.5	19.5	14.4	36.1	26.8
Qwen3.6 35B	129–256	257	219	0.5	0.0	5.5	4.6	63.9	25.6
Qwen3.6 35B	257–512	291	172	0.0	0.6	28.5	11.6	39.0	20.3
Qwen3.6 35B	513–1024	293	90	1.1	6.7	25.6	26.7	4.4	35.6

Table 7: Non-thinking responses shorter than 129 tokens. All main items includes capped and incompletely scored records; eligible full answers requires natural completion and full scoring. Pooled counts sum model–item observations.

	All main items			Eligible full answers		
Model	Items	$L_i < 129$	%	Items	$L_i < 129$	%
All models	14258	6199	43.5	12275	6199	50.5
Gemma-4-31B-it	2376	1238	52.1	2210	1238	56.0
Gemma-4-E4B-it	2377	884	37.2	1902	884	46.5
GLM-4.6V-Flash	2377	1253	52.7	2247	1253	55.8
InternVL3.5-30B-A3B	2375	1408	59.3	2302	1408	61.2
Qwen3.5-4B	2376	707	29.8	1782	707	39.7
Qwen3.6-35B-A3B	2377	709	29.8	1832	709	38.7

D PAIRS CATEGORY AND RELATIVE GAPS

Figure 6 shows every within-model candidate pair on the retained pilot and main-set items, each pooled across the six benchmarks, separately for non-thinking and thinking modes. The horizontal coordinate is the pair mean KLD, $\frac{1}{2}[\overline{D}(\theta_a;T)+\overline{D}(\theta_b;T)]$, and the vertical coordinate is the absolute mean paired difference, $|\widehat{\mu}_{ab}(T)|$, both in nats on logarithmic axes. Pair categories are determined from the pooled pilot items using the multi-cap rule in Section 3.3 and held fixed for both item sets. For KLD computation, we use the token caps specified in Section 5.1: 512 and 2,048 tokens for tight and narrow pairs in non-thinking and thinking modes, respectively, and 256 and 1,024 tokens for moderate and wide pairs. Pilot and main-set estimates are shown in blue and orange, respectively, with two points per pair. All pairs are included regardless of their statistical verdict, and the reference lines show fixed relative gaps at the selected cap.

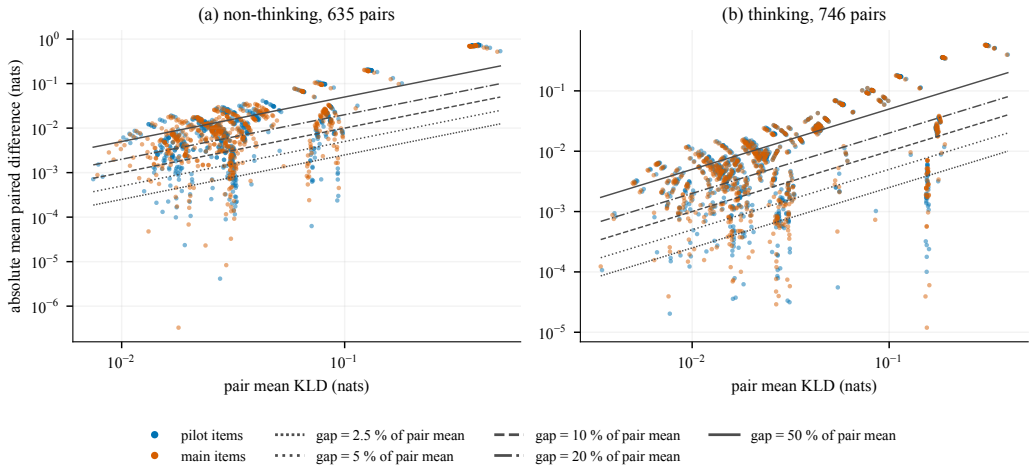

Figure 6: For each of 635 non-thinking and 746 thinking candidate pairs, the absolute mean paired KLD difference is plotted against pair mean KLD on the pooled pilot (blue) and main-set (orange) items at caps selected by the pilot category, with gray lines marking relative gaps of 2.5%, 5%, 10%, 20%, and 50%.

E SNR AND EVALUATION TOKEN CAP ANALYSIS

Apart from the perspectives mentioned in Section 5.1, we also introduce another perspective to select the evaluation token cap. We compare the verdicts from T_{i+1} between T_i, for every resolved verdict at T_{i+1}. Figure 7 repeats the analysis of Section 5.1 but compares each cap T_i with the next cap T_{i+1} instead of T_{limit}; the behavior is similar.

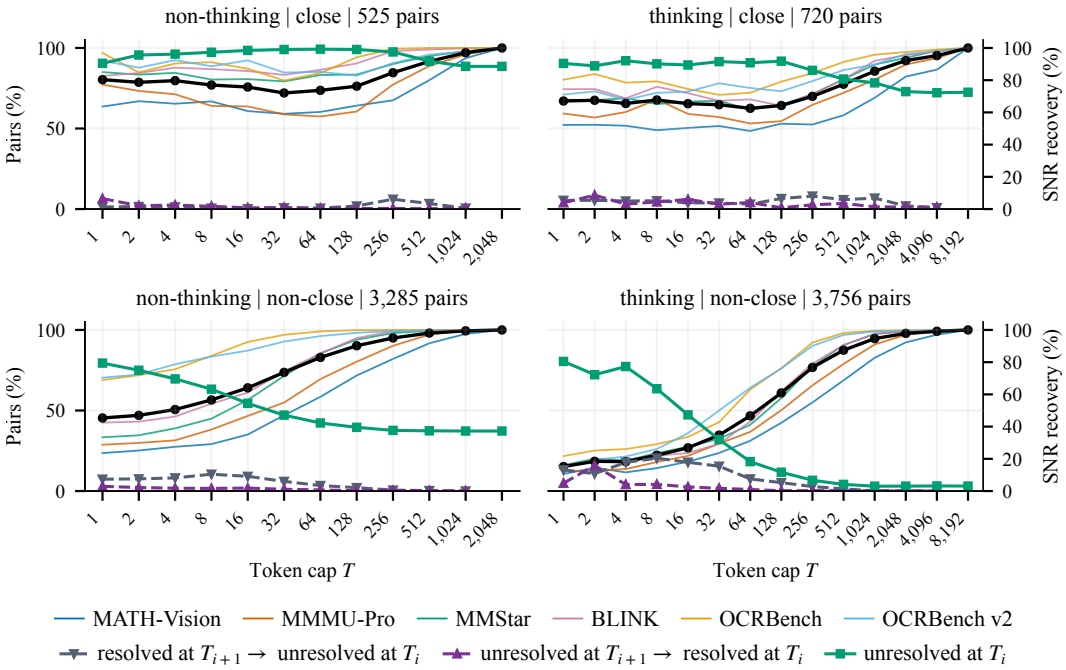

Figure 7: As Figure 1, but each cap T_i is compared with the next cap T_{i+1} instead of T_{limit}: SNR recovery is $|\widehat{\text{SNR}}_{ab}(T_i)|/|\widehat{\text{SNR}}_{ab}(T_{i+1})|$, and the verdict curves give the share of pairs unresolved at T_i and the shares whose verdict changes between T_{i+1} and T_i, with verdicts Holm-corrected within each model–mode–benchmark family at each cap. Pilot items, close and non-close pairs by mode; the thick black line pools models and benchmarks, thin lines are per-benchmark means across models.

In addition, we analyze how often the estimated direction of a pair at a shorter cap disagrees with its verdict at T_{limit}. Figures 8–11 report this rate; their y-axis, *pairs with reversed mean sign vs. T_{limit} (%)*, is defined as follows. For a model and mode, let $\mathcal{R}$ be the set of candidate pairs, of every pair category, whose Holm-corrected verdict (Section 3.3) at T_{limit} is resolved, and let $d_{ab} = \text{sign}\left(\widehat{\mu}_{ab}(T_{\text{limit}})\right) \in \{-1, +1\}$ be the direction of that verdict, where $\widehat{\mu}_{ab}(T)$ is the mean paired difference of Eq. (7) evaluated at cap T. A pair $(\theta_a, \theta_b) \in \mathcal{R}$ has a *reversed mean sign* at cap T if $\text{sign}\left(\widehat{\mu}_{ab}(T)\right) = -d_{ab}$; the verdict at T itself may be resolved or unresolved, and a mean of exactly zero does not count as reversed. The plotted value at cap T is

$$100 \times \frac{\left|\{(\theta_a, \theta_b) \in \mathcal{R} : \text{sign}(\widehat{\mu}_{ab}(T)) = -d_{ab}\}\right|}{|\mathcal{R}|},$$

so pairs that are unresolved at T_{limit} enter neither the numerator nor the denominator, and the value at $T = T_{\text{limit}}$ is zero by construction. Here T_{limit} is the largest evaluated cap of the item set: 2,048/8,192 tokens (non-thinking/thinking) for pilot items and 1,024/2,048 tokens for main items. The rate is computed separately per model and mode over the pooled items (Figures 8 and 9) or per benchmark, with per-benchmark verdicts and pair categories (Figures 10 and 11); a panel reads N/A when no pair is resolved at T_{limit}.

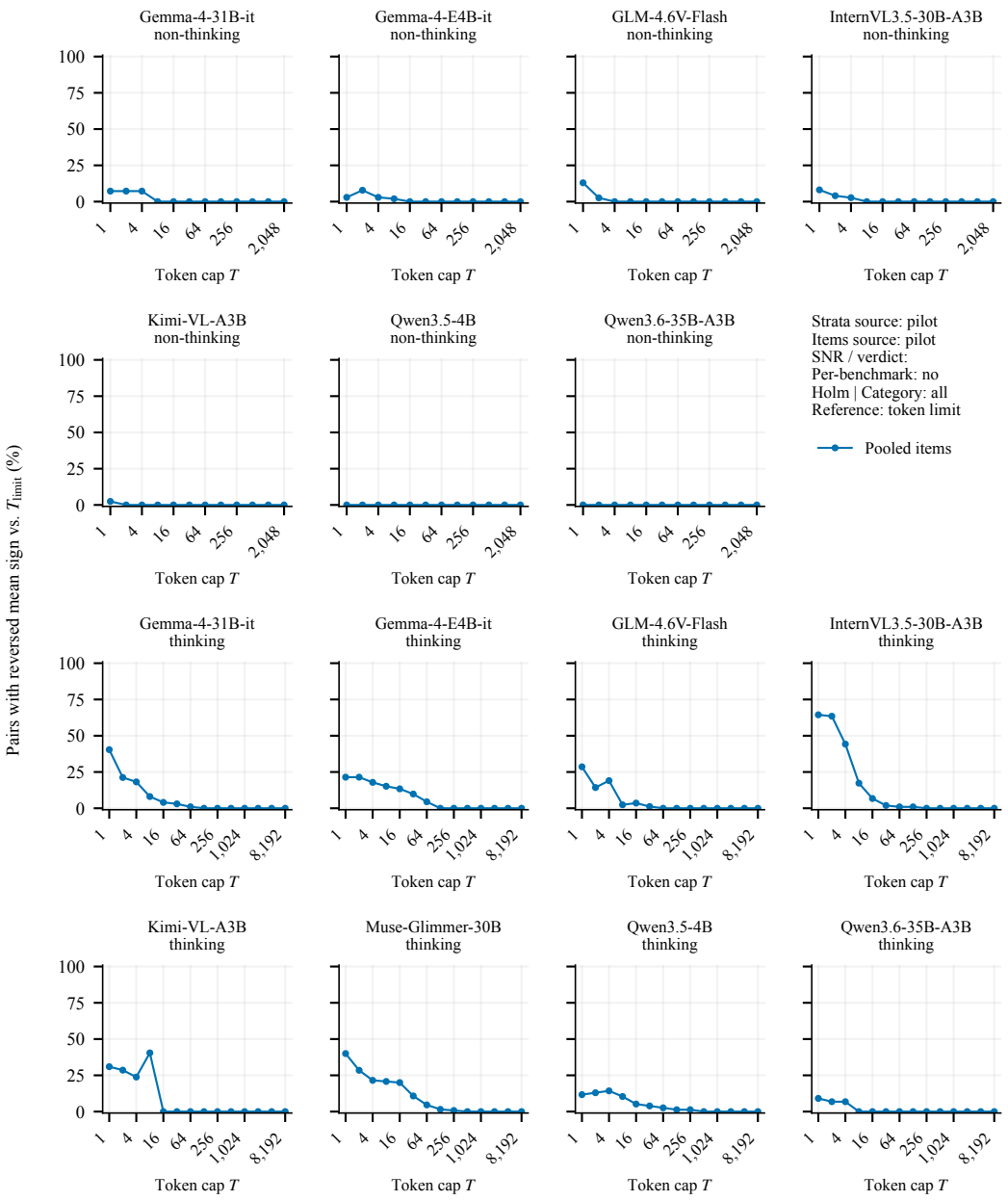

Figure 8: Pairs with reversed mean sign vs. T_{limit} on pilot items with pooled benchmarks. Verdicts at T_{limit} are Holm-corrected within each model–mode family of all candidate pairs.

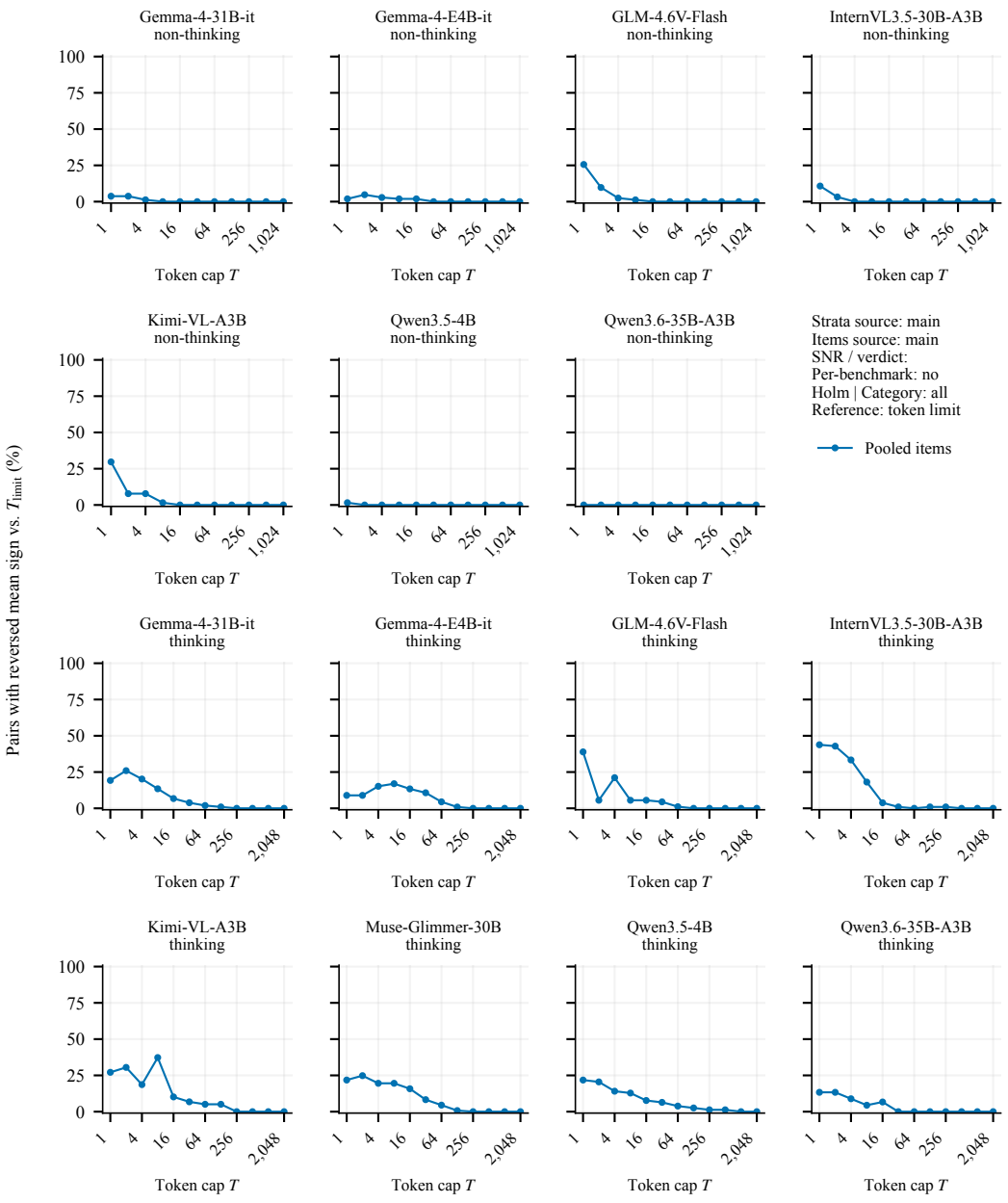

Figure 9: Pairs with reversed mean sign vs. T_{limit} on main items with pooled benchmarks. Verdicts at T_{limit} are Holm-corrected within each model–mode family of all candidate pairs.

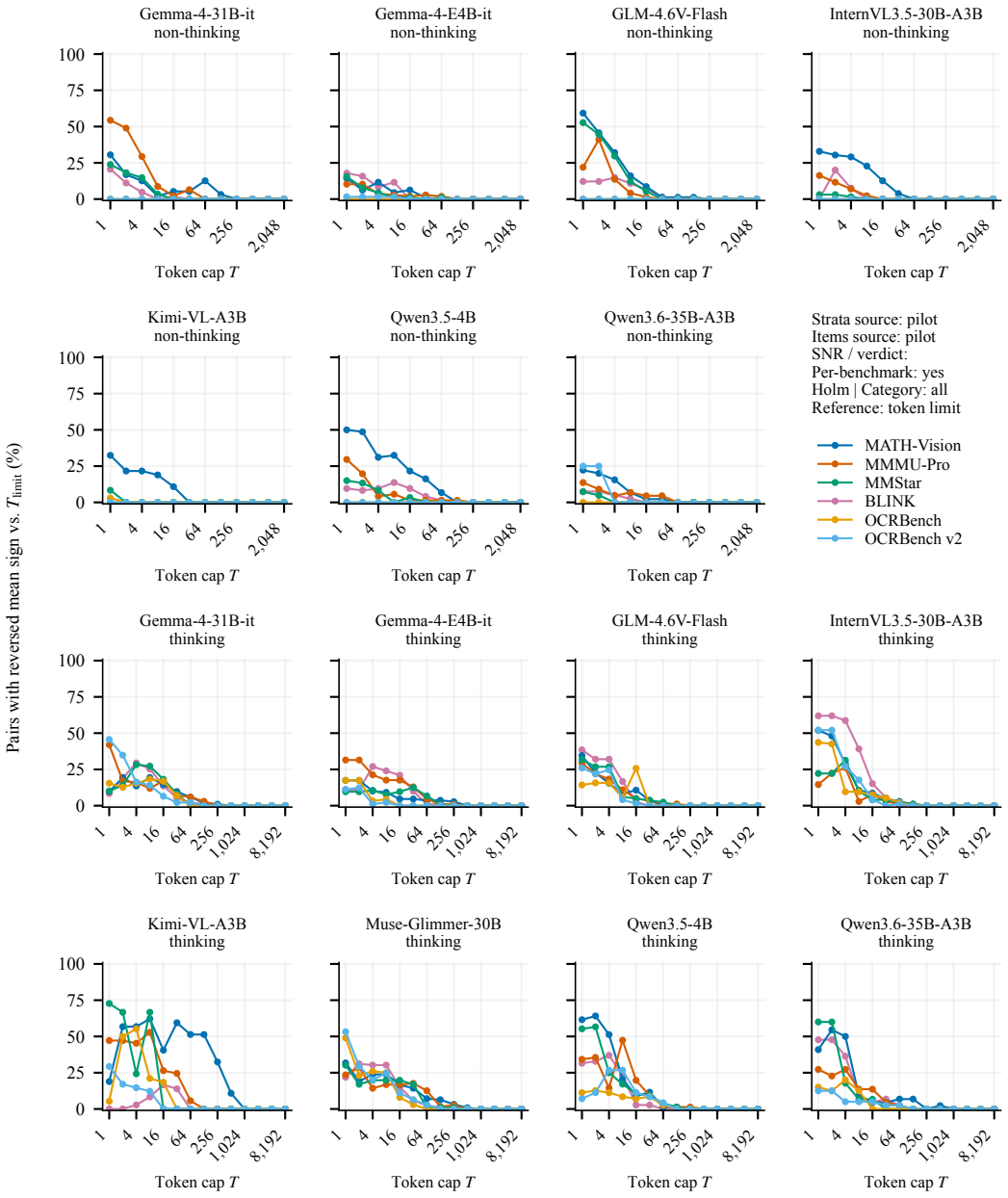

Figure 10: Pairs with reversed mean sign vs. T_{limit} on pilot items with per-benchmark verdicts and pair categories. Verdicts at T_{limit} are Holm-corrected within each model–mode–benchmark family of all candidate pairs.

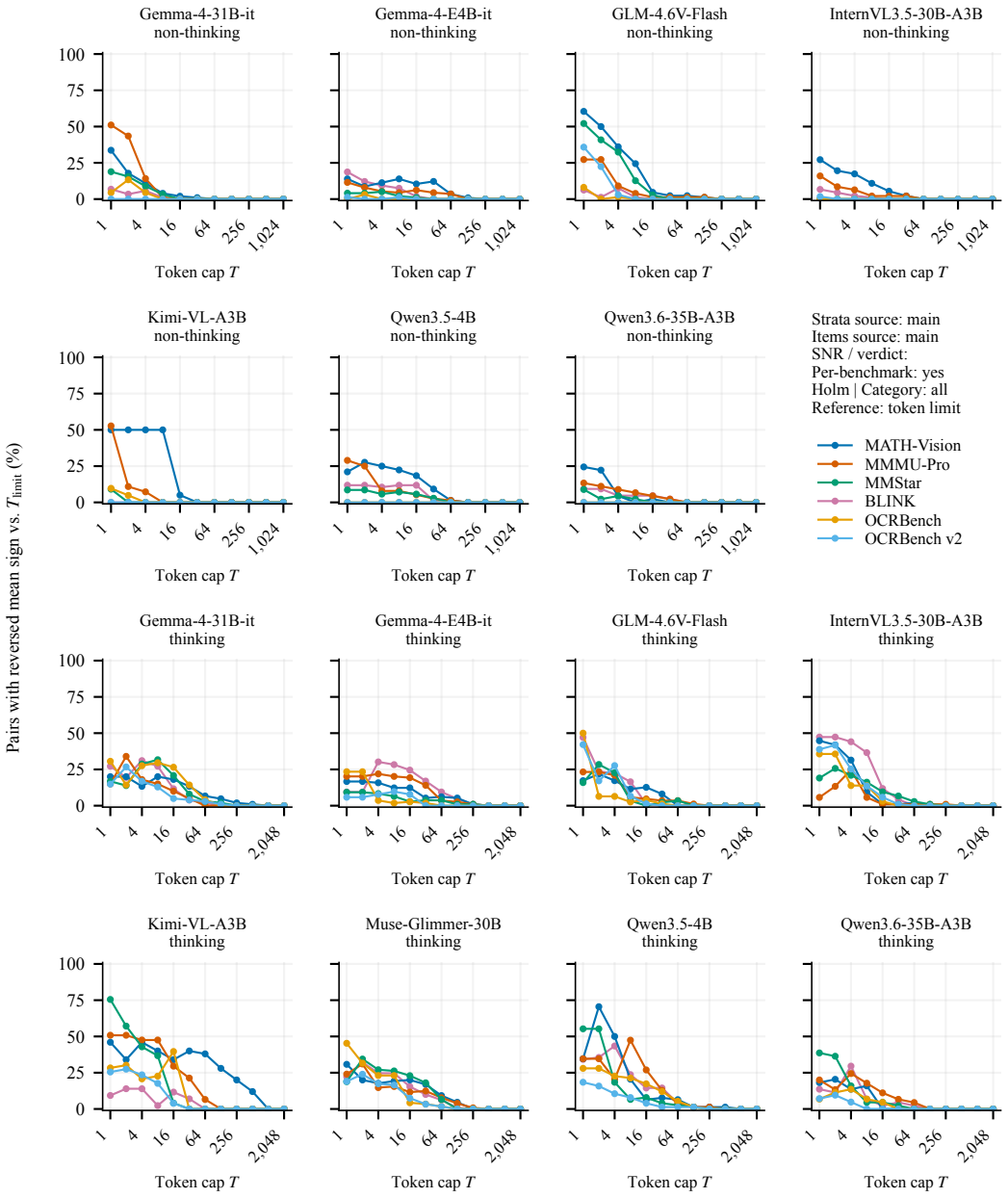

Figure 11: Pairs with reversed mean sign vs. T_{limit} on main items with per-benchmark verdicts and pair categories. Verdicts at T_{limit} are Holm-corrected within each model–mode–benchmark family of all candidate pairs.

F EVALUATE QUANTIZED PAIRS ON DIFFERENT-REFERENCE CORPUS

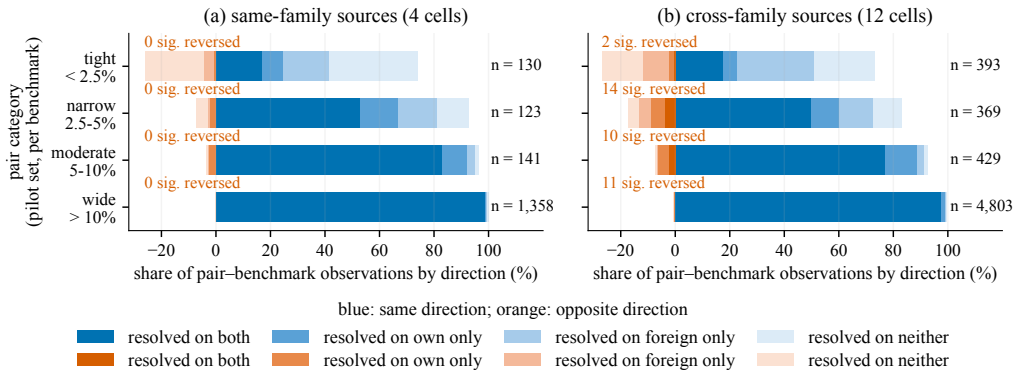

Figure 12: As in Figure 2, but with pair categories determined per benchmark on the pilot set; each observation is therefore a pair–benchmark combination rather than a pair.

G QUANTIZATION ON DUAL MODE CHECKPOINTS

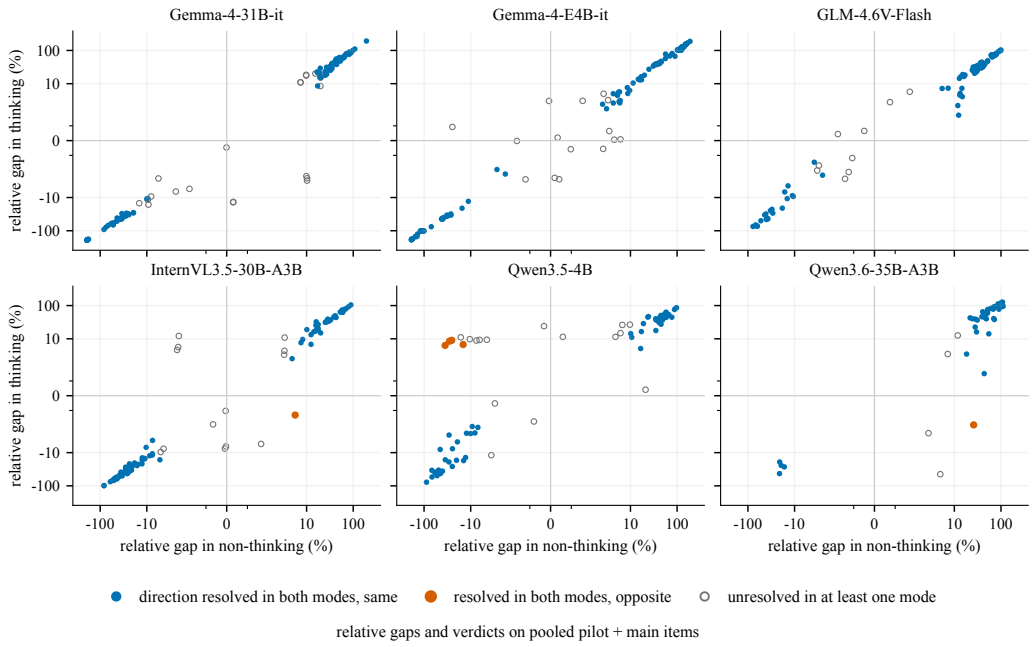

Figure 13: Quantization on dual-mode models on the pilot-plus-main set. We use evaluation token cap 512/2,048 for non-thinking and thinking mode as we used in the main text. The verdicts are Holm-corrected.

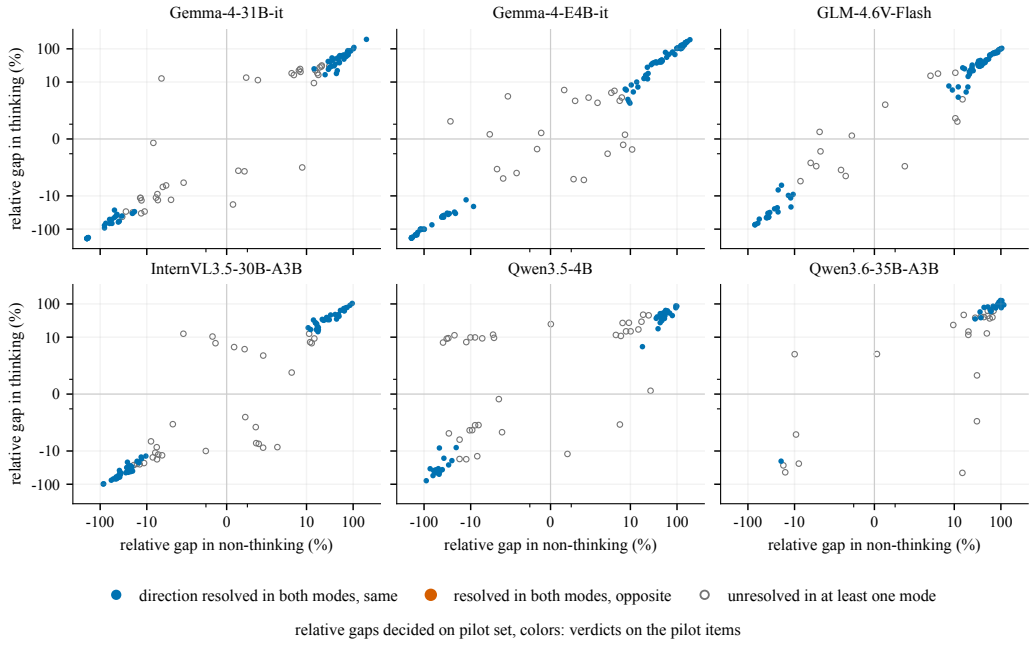

Figure 14: Quantization on dual-mode models on the pilot set. We use evaluation token cap 512/2,048 for non-thinking and thinking. The verdicts are Holm-corrected.

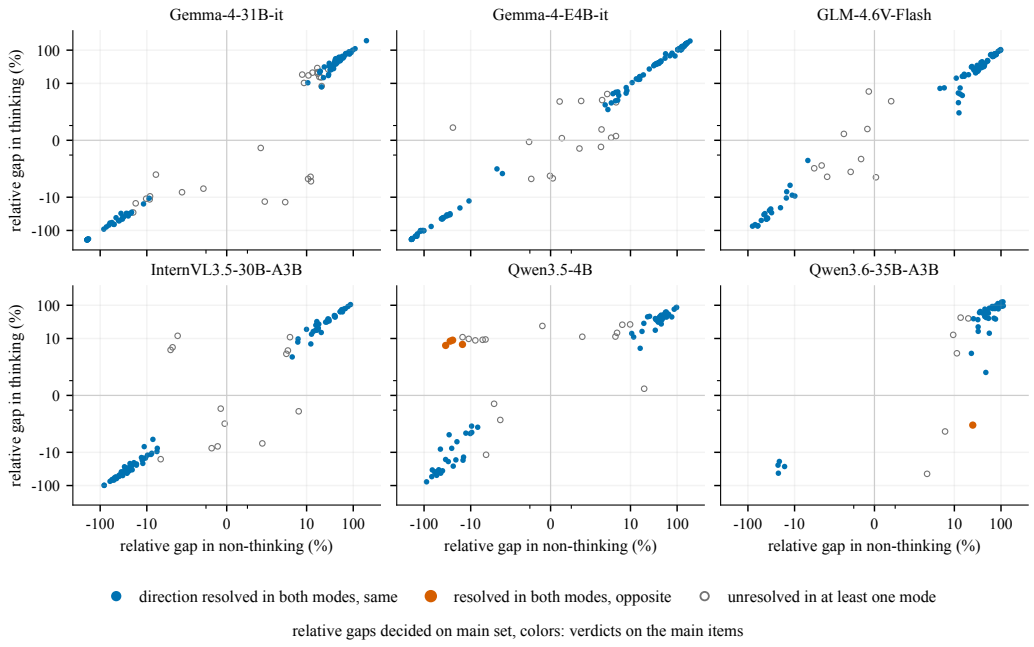

Figure 15: Quantization on dual-mode models on the main set. We use evaluation token cap 512/2,048 for non-thinking and thinking mode as we used in the main text. The verdicts are Holm-corrected.

Table 8: Candidate pairs shown as orange points in Figures 13–15; a checkmark indicates opposite directions resolved in both modes after Holm correction ($p < 0.05$ in each), and a dash indicates otherwise. Q3.5, Q3.6, and IVL3.5 denote Qwen3.5-4B, Qwen3.6-35B-A3B, and InternVL3.5-30B-A3B, respectively. Prefixes bar-, un-, and mra- denote bartowski, unsloth, and mradermacher, respectively.

Candidate A	Candidate B	Fig. 13	Fig. 14	Fig. 15
Q3.5				
bar-IQ4_NL	un-Q4_1	✓	—	✓
bar-IQ4_NL	un-Q4_K_S	✓	—	✓
bar-IQ4_XS	un-Q4_1	✓	—	✓
un-IQ4_NL	un-Q4_1	✓	—	✓
Q3.6				
bar-IQ4_NL	bar-IQ4_XS	✓	—	✓
IVL3.5				
mra-IQ4_XS	mra-Q4_K_M	✓	—	—

Table 9: Holm-corrected p-values for the pairs in Table 8, adjusting all pairs in both modes within each model and item set. Model and candidate abbreviations follow Table 8. NT/T denotes non-thinking/thinking; bold values indicate $p \geq 0.05$.

		Fig. 13		Fig. 14		Fig. 15	
Candidate A	Candidate B	NT	T	NT	T	NT	T
Q3.5							
bar-IQ4_NL	un-Q4_1	$< 10^{-5}$	$< 10^{-5}$	**0.2251**	$< 10^{-5}$	1.54×10^{-5}	$< 10^{-5}$
bar-IQ4_NL	un-Q4_K_S	0.0015	$< 10^{-5}$	**1.0000**	$< 10^{-5}$	0.0066	$< 10^{-5}$
bar-IQ4_XS	un-Q4_1	0.0001	$< 10^{-5}$	**0.4599**	$< 10^{-5}$	0.0010	$< 10^{-5}$
un-IQ4_NL	un-Q4_1	0.0003	$< 10^{-5}$	**0.5700**	$< 10^{-5}$	0.0036	$< 10^{-5}$
Q3.6							
bar-IQ4_NL	bar-IQ4_XS	0.0026	$< 10^{-5}$	**1.0000**	0.0002	0.0068	$< 10^{-5}$
IVL3.5							
mra-IQ4_XS	mra-Q4_K_M	0.0010	0.0064	**1.0000**	0.0178	0.0003	**0.1434**

H SELECTED BENCHMARKS FOR VLMS

Our benchmark selection is guided by two considerations: coverage of different VLM capabilities, and an emphasis on questions that cannot be answered without the image.

MMMU-Pro. MMMU-Pro (Yue et al., 2025) is a more challenging variant of the Massive Multi-discipline Multimodal Understanding and Reasoning (MMMU) benchmark (Yue et al., 2024), built by filtering out questions answerable by text-only LLMs, expanding the candidate options from four to up to ten, and adding a vision-only input setting. The benchmark ships three subsets: standard (4 options), standard (10 options), and vision. We analyze only the vision subset; standard (10 options) was collected separately but excluded because it shares questions with the vision variant. In the vision subset, the question, answer options, and associated images are rendered together in a single screenshot or photo, so the model must read and interpret textual and visual information jointly from the image without receiving the question as separate text input.

MATH-Vision. MATH-Vision (Wang et al., 2024b) is a benchmark for multimodal mathematical reasoning comprising 3,040 problems sourced from 19 mathematics competitions. The problems span 16 mathematical subjects and five difficulty levels, with a nearly even split between open-ended and multiple-choice formats. In addition to the full test set, MATH-Vision provides a 304-problem testmini subset for quick evaluation. In our work, we sample questions from the test set directly.

MMStar. MMStar (Chen et al., 2024) comprises 1,500 multiple-choice questions automatically filtered from existing benchmarks and manually verified to require visual understanding, organized into six core capabilities and 18 detailed axes. Its questions are drawn from SEED-Bench (28.3%), MMBench (24.4%), MathVista (23.0%), AI2D (13.1%), MMMU (6.7%), and ScienceQA (4.6%). We select MMStar for its coverage of diverse capabilities and its continued use in the research community.

OCRBench (v1). OCRBench (Liu et al., 2024) evaluates the text-reading ability of VLMs with 1,000 question–answer pairs drawn from 29 datasets, grouped into five task types: text recognition, scene text-centric VQA, document-oriented VQA, key information extraction, and handwritten mathematical expression recognition. We include OCRBench to cover text-centric inputs, whose responses differ in character from those of general VQA.

OCRBench v2. OCRBench v2 (Fu et al., 2025) expands the scope of OCRBench (Liu et al., 2024), increasing the benchmark from 1,000 to 10,000 human-verified question–answer pairs, with 23 tasks grouped into eight core capabilities across 31 scenarios, covering both English and Chinese instructions represented across the benchmark. Beyond text recognition, models must localize text in images, map information to the correct fields, convert tables, charts, and documents into structured formats, and answer questions requiring mathematical, visual, or knowledge-based reasoning over image-embedded text.

BLINK. BLINK (Fu et al., 2024) evaluates visual perception abilities through questions that humans can generally answer at a glance but that remain challenging for contemporary VLMs. It contains 3,807 multiple-choice questions over 7,358 images and covers 14 tasks drawn from classic computer vision problems and recent multimodal-model applications, ranging from low-level pattern matching through mid-level spatial reasoning to high-level visual understanding.

I DATASET RETENTION AFTER FILTERING

Table 10: Item counts after shared-image cleaning

Benchmark	Pilot	Main	Total	Removed
MATH-Vision	100	400	500	0
MMMU-Pro (vision)	100	400	500	0
MMStar	99	387	486	14
BLINK	100	399	499	1
OCRBench v1	94	394	488	12
OCRBench v2	100	398	498	2
Total	593	2378	2971	29

The six analyzed benchmarks have a nominal budget of 3,000 items. Table 10 describes the number of samples after shared-image cleaning but before reference generation. The cleaning removes 29 items whose images are shared across benchmarks.

Then, we use the image-conditioned question set with 2,971 items to build the reference dataset for each model-mode. We use a repetition detector and Fable 5.1 or GPT-6-Astra to clean the repetition cases. Table 11 shows the number of retention samples in each model-mode.

Table 11: Item counts after reference generation and repetition filtering, per benchmark, model and mode

Model	MATH Vision	MMMU-Pro (vision)	MMStar	BLINK	OCRBench v1	OCRBench v2	All
				Pilot			
Instruct							
Gemma 4 31B	100	100	99	100	94	100	593
Gemma 4 E4B	100	100	99	100	94	100	593
GLM 4.6V Flash	100	100	99	100	94	100	593
InternVL 3.5 30B	99	100	98	100	94	100	591
Kimi VL A3B	99	81	99	100	89	90	558
Muse Glimmer 30B	–	–	–	–	–	–	–
Qwen 3.5 4B	100	100	99	100	94	100	593
Qwen 3.6 35B A3B	100	100	99	100	94	100	593
Thinking							
Gemma 4 31B	100	100	99	100	94	99	592
Gemma 4 E4B	100	100	99	100	94	100	593
GLM 4.6V Flash	100	100	99	100	94	100	593
InternVL 3.5 30B	99	98	99	100	94	100	590
Kimi VL A3B	98	95	99	100	92	88	572
Muse Glimmer 30B	99	100	99	100	94	100	592
Qwen 3.5 4B	100	100	99	100	94	100	593
Qwen 3.6 35B A3B	100	100	99	100	94	100	593
				Main			
Instruct							
Gemma 4 31B	400	400	387	399	394	396	2376
Gemma 4 E4B	400	400	387	399	394	397	2377
GLM 4.6V Flash	400	400	387	399	394	397	2377
InternVL 3.5 30B	400	400	387	399	394	395	2375
Kimi VL A3B	387	312	381	399	385	340	2204
Muse Glimmer 30B	–	–	–	–	–	–	–
Qwen 3.5 4B	400	400	387	399	394	396	2376
Qwen 3.6 35B A3B	400	400	387	399	394	397	2377
Thinking							
Gemma 4 31B	400	399	387	399	394	396	2375
Gemma 4 E4B	400	400	387	399	394	397	2377
GLM 4.6V Flash	400	400	387	399	394	395	2375
InternVL 3.5 30B	399	398	387	399	393	395	2371
Kimi VL A3B	396	379	385	396	391	355	2302
Muse Glimmer 30B	400	400	387	399	394	397	2377
Qwen 3.5 4B	400	400	387	399	394	396	2376
Qwen 3.6 35B A3B	399	400	387	399	394	396	2375

J REFERENCE DATASET GENERATION

J.1 REFERENCE MODEL CHECKPOINTS

Table 12: The models used to generate the reference datasets

Model hub	**Version**	**Size**	**Arch.**
bartowski/google_gemma-4-E4B-it-GGUF	029e94146	7.52B	Dense
bartowski/Qwen_Qwen3.5-4B-GGUF	4168f45a1	4.21B	Dense
Kimi-VL-A3B-Thinking-2506 (llama.cpp)	6a1a922d2	15.96B	MoE
Kimi-VL-A3B-Instruct (llama.cpp)	6a1a922d2	15.96B	MoE
bartowski/google_gemma-4-31B-it-GGUF	619aca758	30.70B	Dense
bartowski/Qwen_Qwen3.6-35B-A3B-GGUF	5c2410d71	34.66B	MoE
bartowski/Muse-Glimmer-30B-GGUF	01e1b7c30	27.85B	Dense
bartowski/OpenGVLab_InternVL3_5-30B-A3B-GGUF	a6824f88f	30.53B	MoE
bartowski/zai-org_GLM-4.6V-Flash-GGUF	6a3b3ca7b	9.40B	Dense

We use the BF16 LLM and BF16 mmproj GGUF from bartowski as our reference models. For the Kimi-VL-A3B series, we could not find the corresponding checkpoints from bartowski, so we convert the model from Hugging Face with llama.cpp directly with the upstream commit 6a1a922d2.

J.2 PERCENTAGE OF THE RESPONSE LENGTHS IN PILOT THAT EXCEED THE EVALUATION LIMIT TOKEN CAP

Table 13: Percentage of pilot reference responses that are longer than the evaluation limit (2,048 tokens for non-thinking, 8,192 for thinking), per benchmark and over the six benchmarks (All), with the count k of such responses over the N retained pilot items of the six benchmarks.

	Non-Thinking (limit 2,048)							
Model	MATH Vision	MMMU-Pro vis	MMStar	BLINK	OCRBench v1	OCRBench v2	All	k/N
Gemma 4 31B	8.0	1.0	0.0	0.0	0.0	0.0	1.5	9/593
Gemma 4 E4B	43.0	12.0	1.0	0.0	0.0	0.0	9.4	56/593
GLM 4.6V Flash	3.0	2.0	0.0	0.0	0.0	0.0	0.8	5/593
InternVL 3.5 30B	3.0	1.0	0.0	0.0	0.0	0.0	0.7	4/591
Kimi VL A3B	4.0	3.7	0.0	0.0	0.0	4.4	2.0	11/558
Muse Glimmer 30B	–	–	–	–	–	–	–	–
Qwen 3.5 4B	73.0	39.0	8.1	4.0	0.0	1.0	21.1	125/593
Qwen 3.6 35B A3B	64.0	34.0	7.1	1.0	0.0	1.0	18.0	107/593
All items	28.4	13.5	2.3	0.7	0.0	0.9	7.7	317/4114
	Thinking (limit 8,192)							
Model	MATH Vision	MMMU-Pro vis	MMStar	BLINK	OCRBench v1	OCRBench v2	All	k/N
Gemma 4 31B	30.0	11.0	4.0	3.0	0.0	1.0	8.3	49/592
Gemma 4 E4B	3.0	1.0	0.0	0.0	0.0	0.0	0.7	4/593
GLM 4.6V Flash	19.0	7.0	0.0	0.0	0.0	0.0	4.4	26/593
InternVL 3.5 30B	31.3	17.3	2.0	1.0	0.0	0.0	8.6	51/590
Kimi VL A3B	28.6	4.2	0.0	0.0	0.0	0.0	5.6	32/572
Muse Glimmer 30B	31.3	8.0	2.0	2.0	0.0	2.0	7.6	45/592
Qwen 3.5 4B	82.0	60.0	22.2	28.0	9.6	19.0	37.1	220/593
Qwen 3.6 35B A3B	69.0	50.0	23.2	18.0	1.1	9.0	28.7	170/593
All items	36.8	19.9	6.7	6.5	1.3	3.9	12.7	597/4718

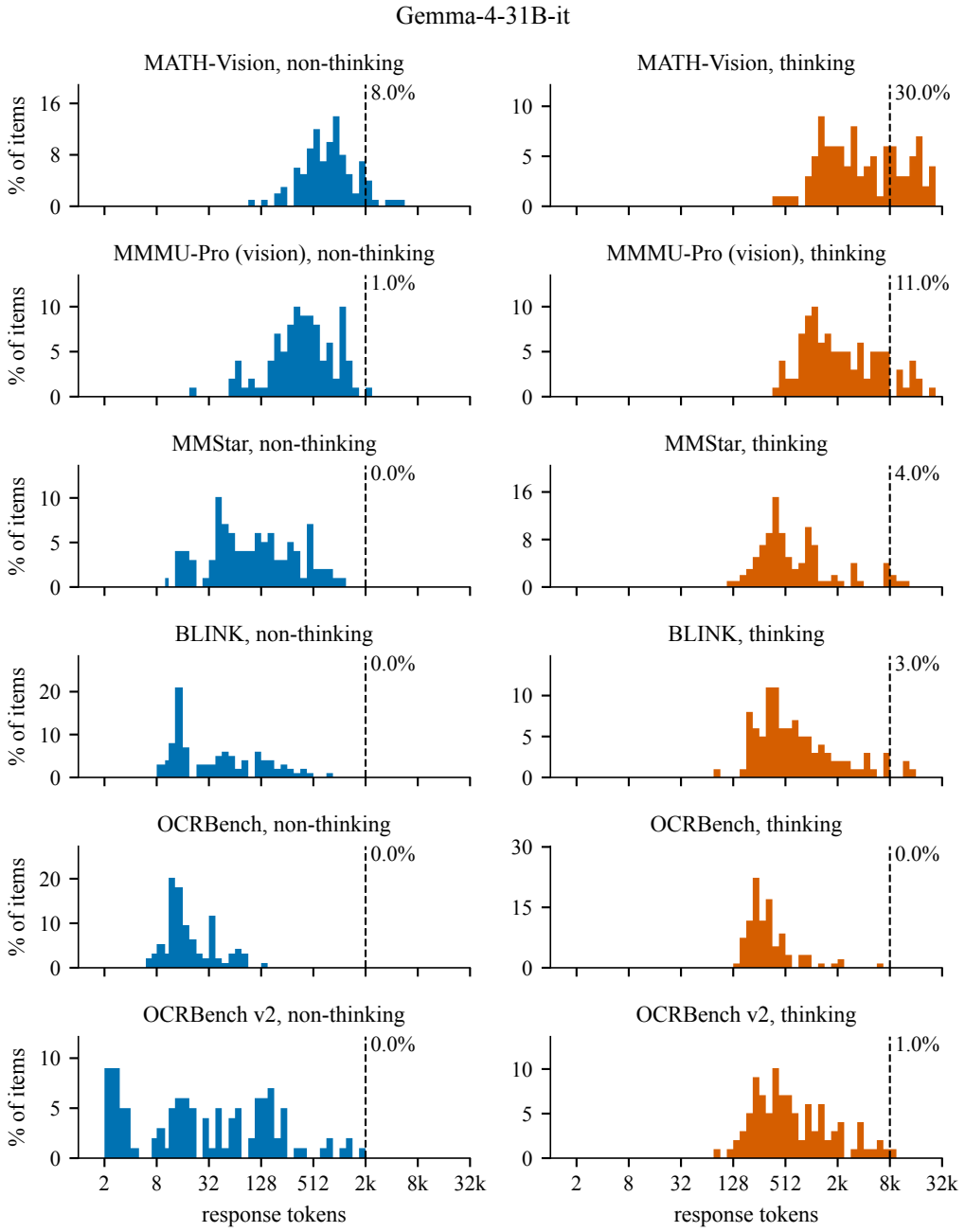

Figure 16: Histogram of response lengths in the pilot set of Gemma-4-31B-it. Rows are the six benchmarks and columns the non-thinking and thinking modes. The x axis is the number of generated tokens (the thinking trace included) on a $\log_2$ scale in quarter-octave bins (k = 1,024); the y axis is the percentage of the benchmark's retained pilot items. The dashed line marks the evaluation limit (2,048 tokens for non-thinking, 8,192 for thinking), and the number to its right is the percentage of responses longer than the limit (Table 13).

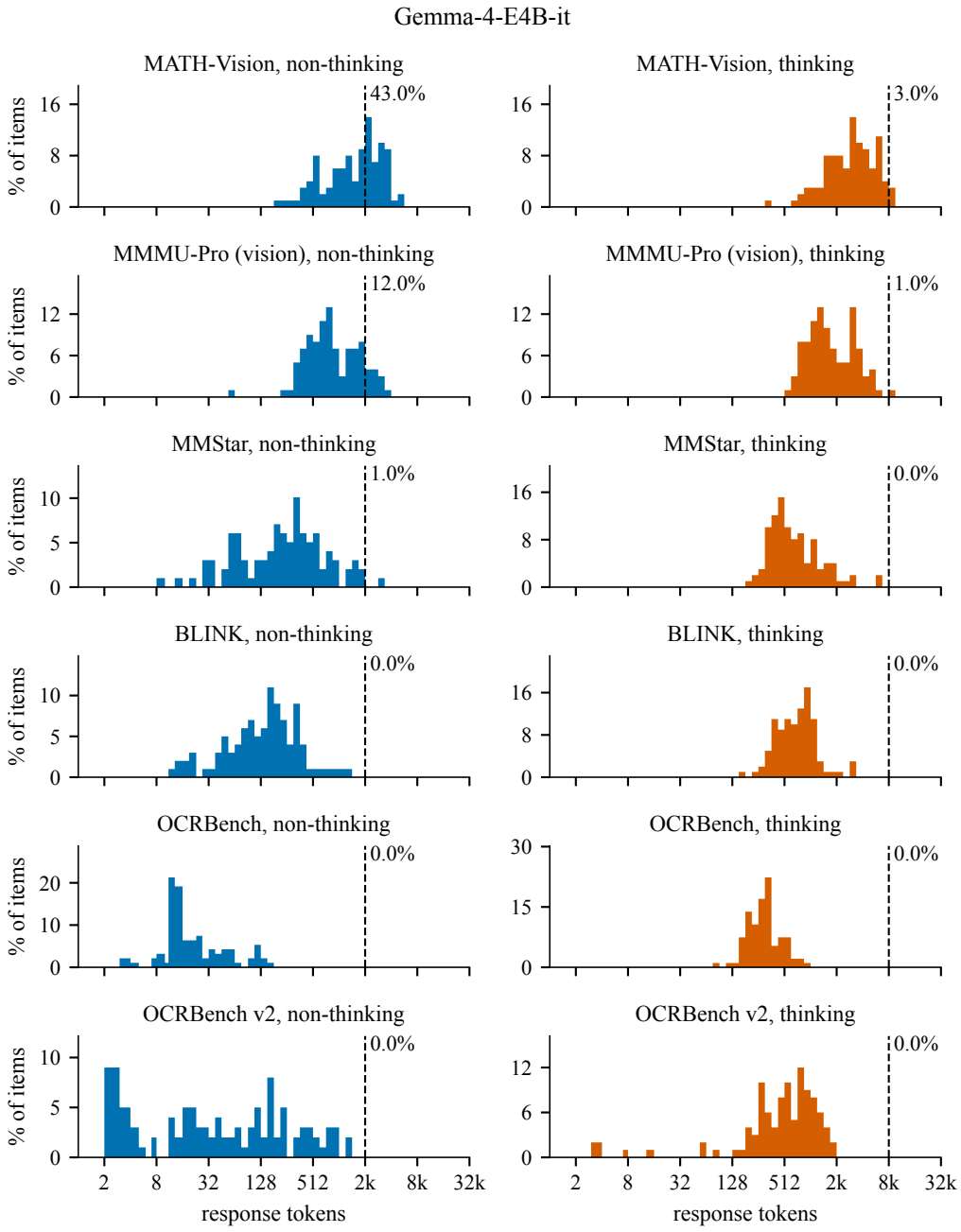

Figure 17: Histogram of response lengths in the pilot set of Gemma-4-E4B-it, as in Figure 16.

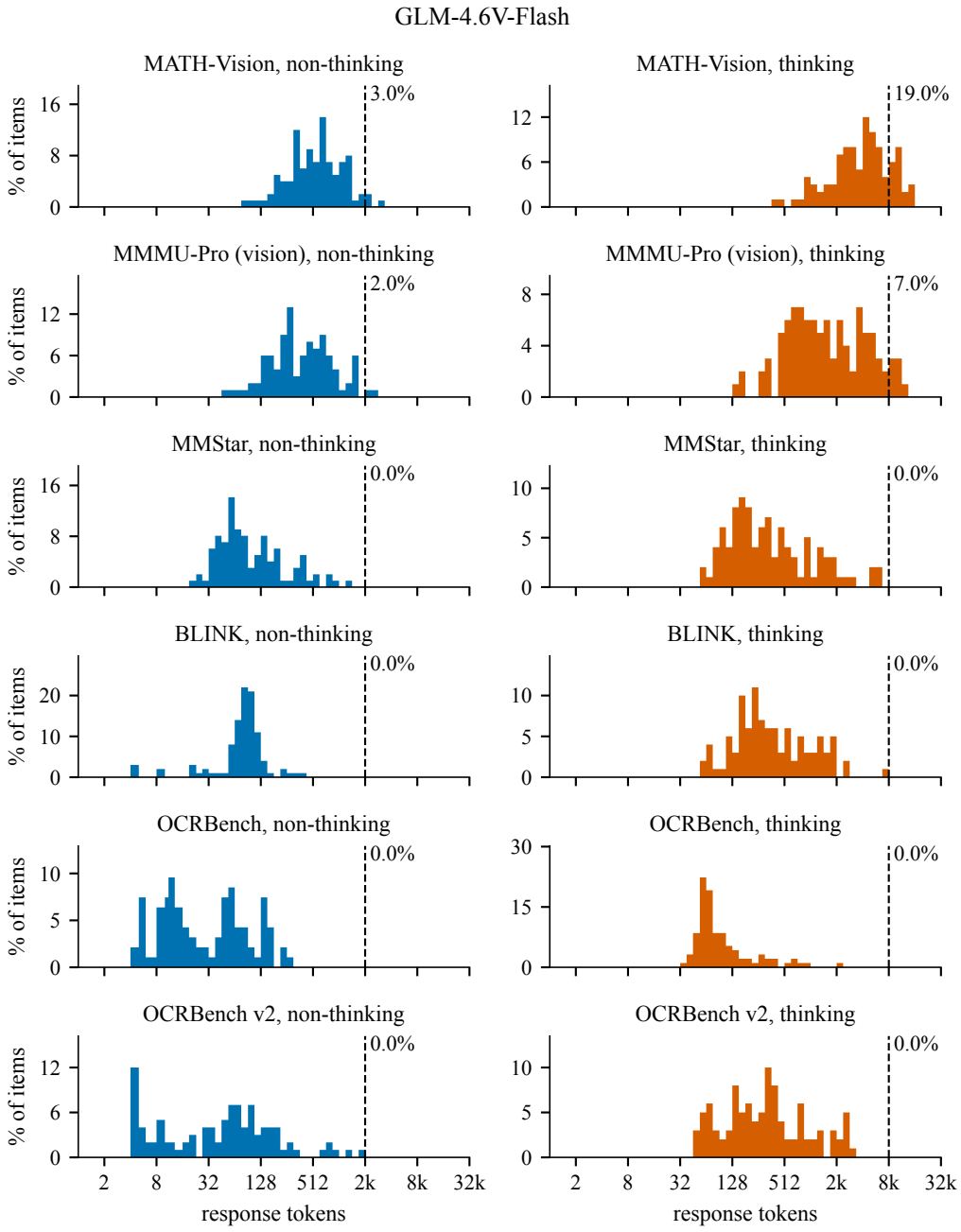

Figure 18: Histogram of response lengths in the pilot set of GLM-4.6V-Flash, as in Figure 16.

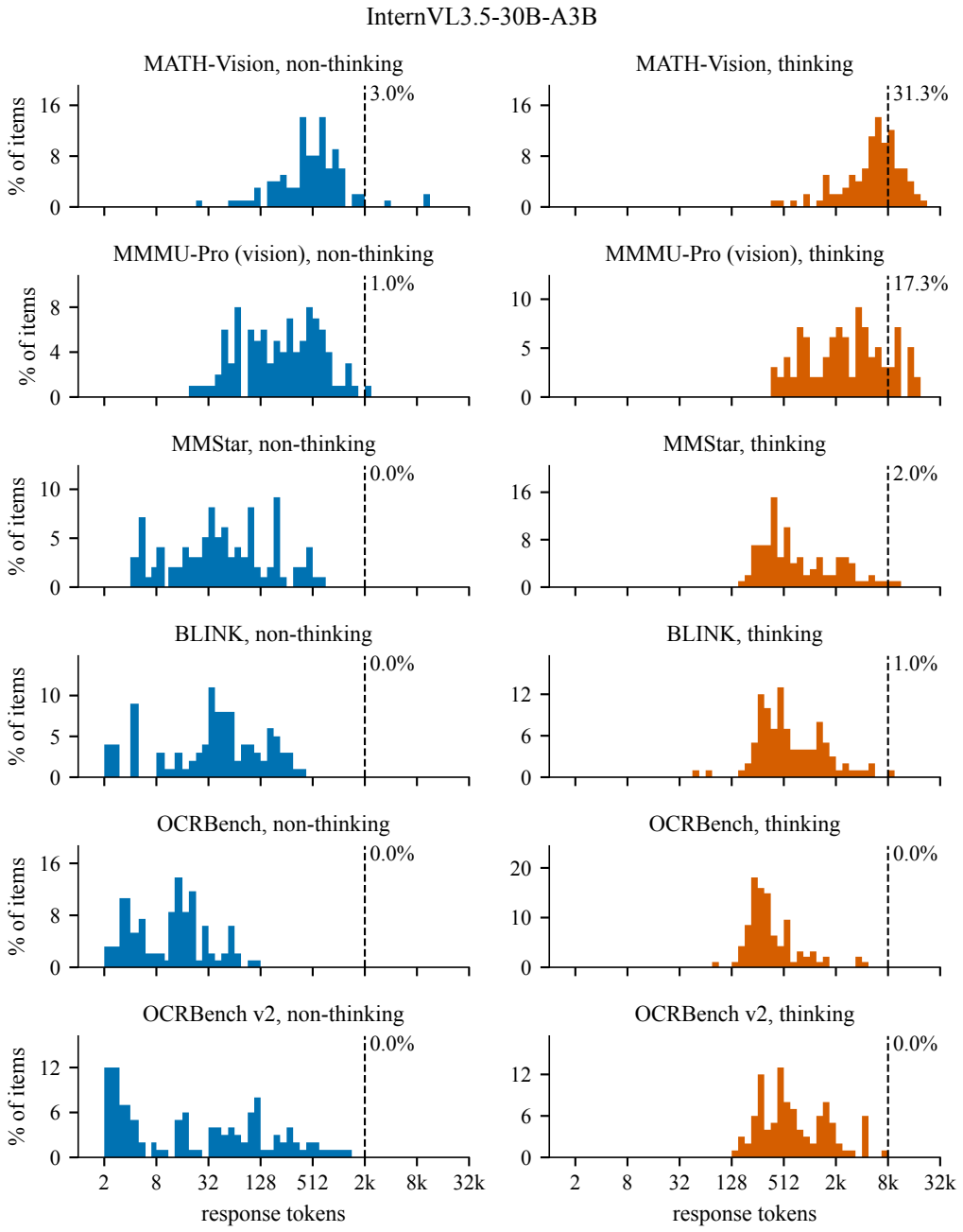

Figure 19: Histogram of response lengths in the pilot set of InternVL3.5-30B-A3B, as in Figure 16.

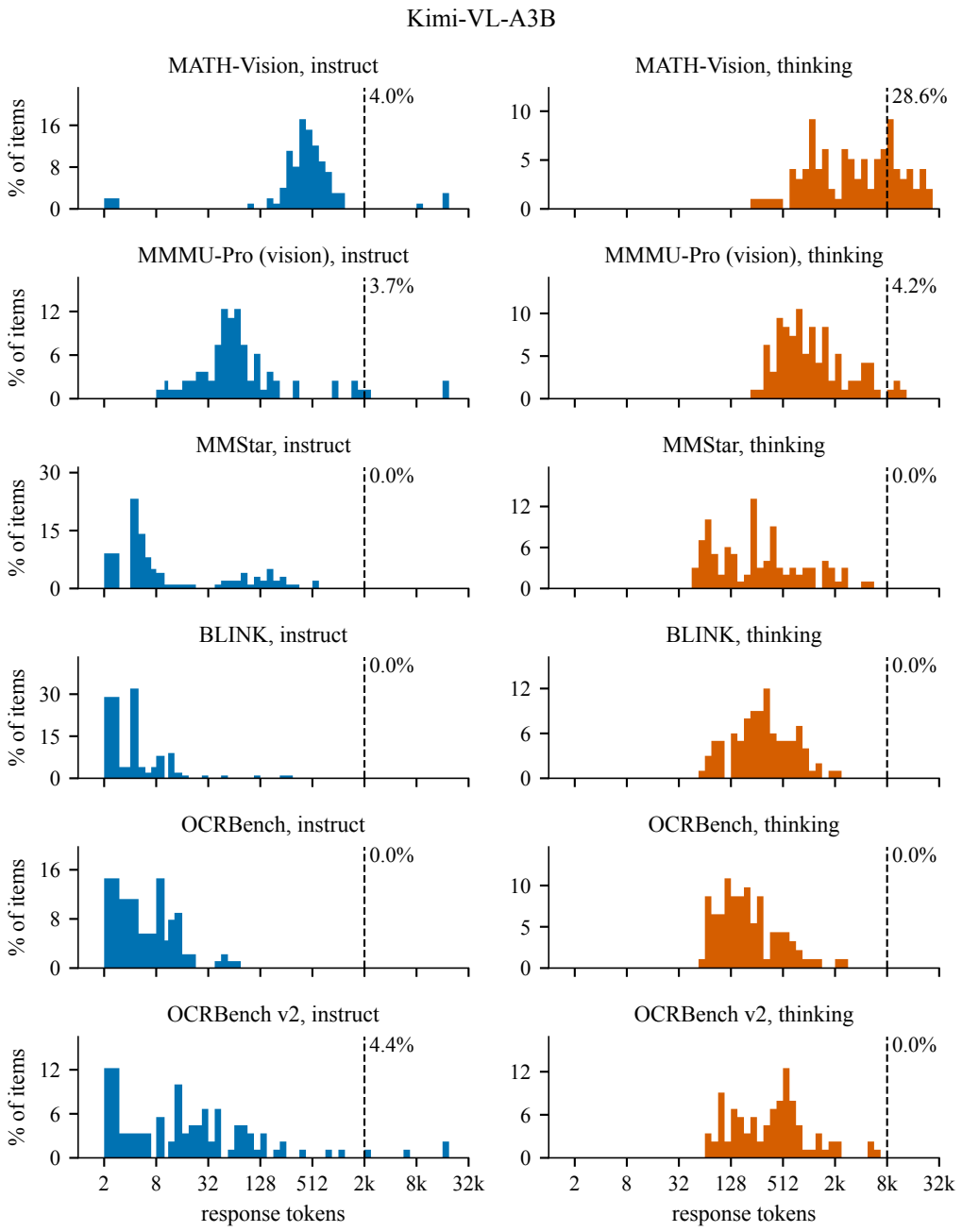

Figure 20: Histogram of response lengths in the pilot set of Kimi-VL-A3B, as in Figure 16: instruct from Kimi-VL-A3B-Instruct, thinking from Kimi-VL-A3B-Thinking-2506.

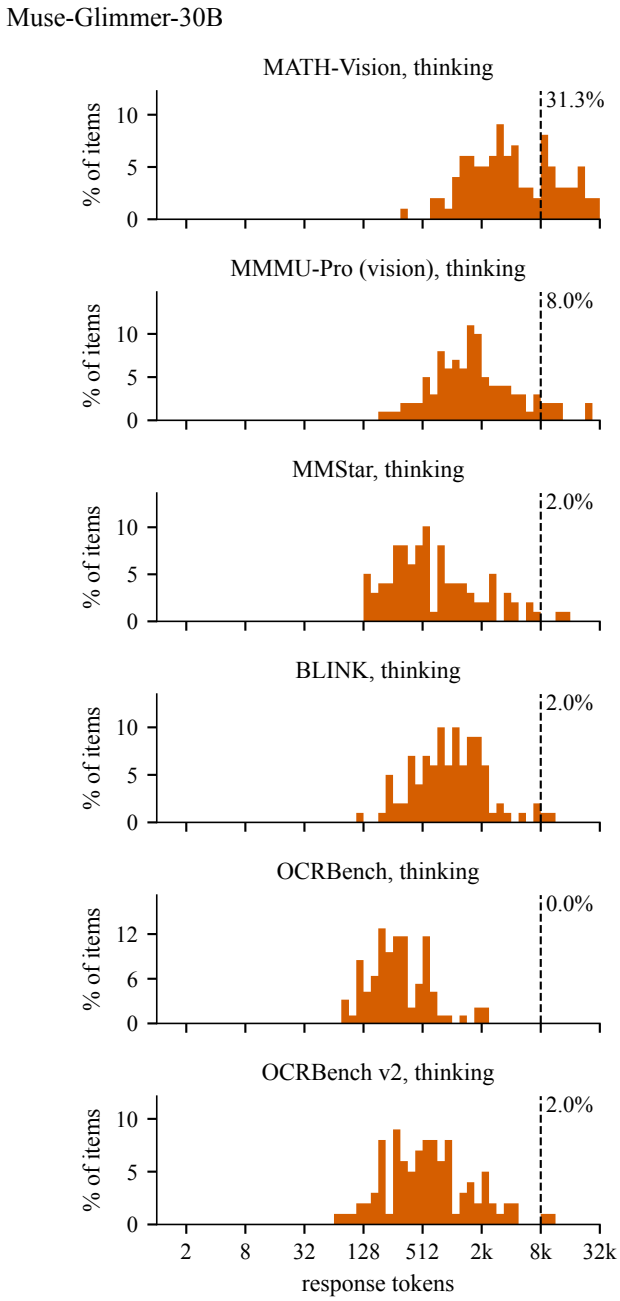

Figure 21: Histogram of response lengths in the pilot set of Muse-Glimmer-30B, as in Figure 16; the model has no non-thinking mode.

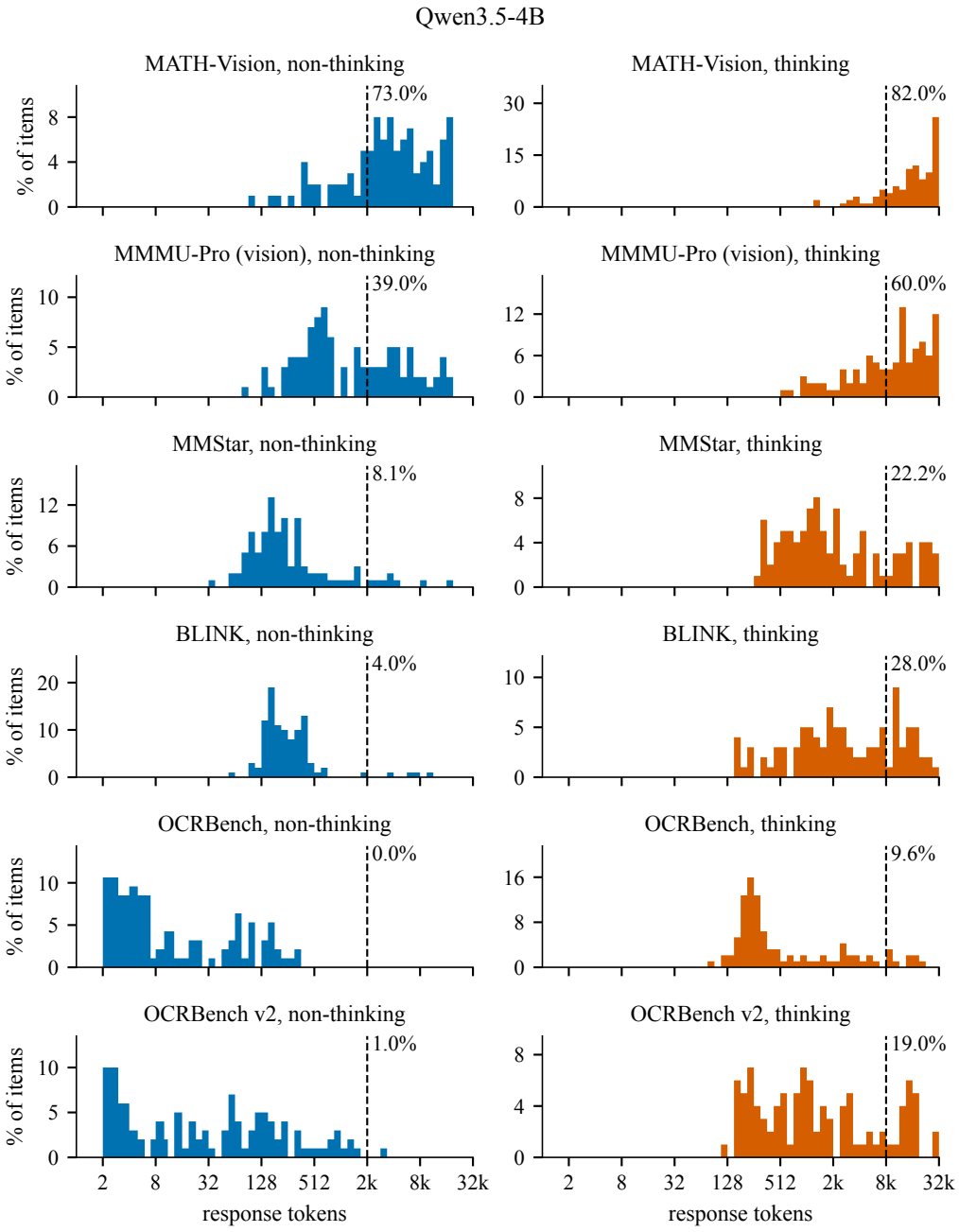

Figure 22: Histogram of response lengths in the pilot set of Qwen3.5-4B, as in Figure 16.

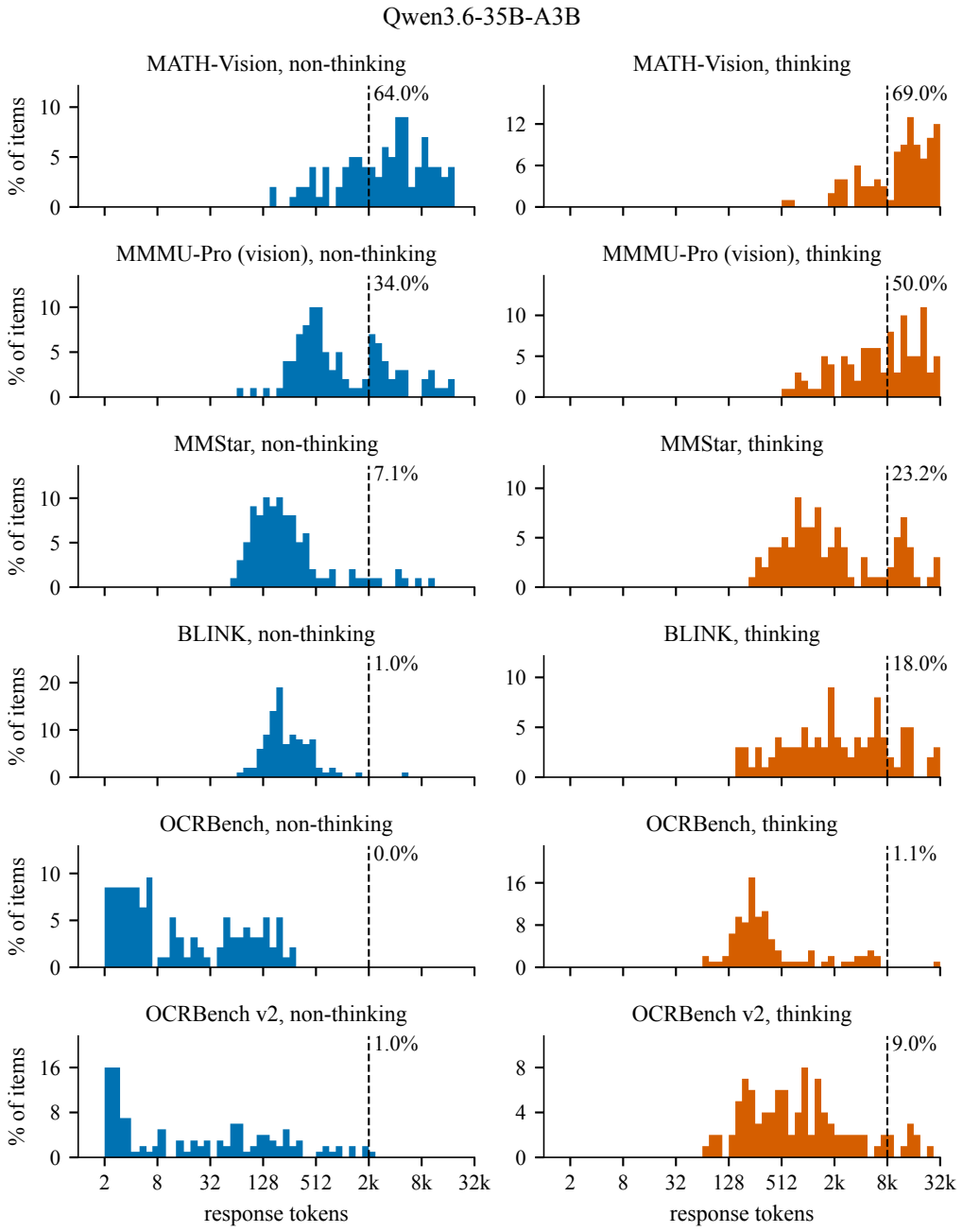

Figure 23: Histogram of response lengths in the pilot set of Qwen3.6-35B-A3B, as in Figure 16.

J.3 Runtime Settings for Reference generation

Table 14 lists the runtime of the customized llama-reference binary used to generate the reference responses.

Table 14: Runtime settings of reference response generation (customized llama-reference binary)

Setting	Value	Notes
Context length	49,152	
Logical batch / micro-batch	2,048 / 512	micro-batch 2,048 for Gemma-4-31B-it
Threads (decode / batch)	8 / 8	
Parallel sequences	1	4 for Qwen3.5-4B and Gemma-4-E4B-it
Speculative draft (`draft_max`)	0	8 for Gemma-4-31B-it, drafting from its Q8_0 multi-token-prediction sidecar head
GPU offload	all layers	
Flash attention	on	
KV cache type (K / V)	f16 / f16	
Context auto-fit (`--fit`)	off	
SWA full cache	off	
Seed	1234	
Image token budget	native	`--image-min-tokens` / `--image-max-tokens` omitted: the GGUF/mtmd defaults of each model
Generation cap (tokens)	16,384 / 32,768	non-thinking / thinking; main set 1,024 / 4,096
Sampling	Table 15	model-card parameters in llama.cpp's native sampler order
Hardware	1 GPU per run	NVIDIA RTX PRO 6000 Blackwell Server Edition

Values are the frozen per-model generation profiles of the pilot set; every model uses the same profile in both modes. The responses in the pilot dataset were generated with these settings and the main set uses the same configurations except for the generation cap.

J.4 Decoding Parameters

Table 15: Decoding parameters to generate reference dataset

Model	**Mode**	**Temp.**	**Top-k**	**Top-p**	$p_{\min}$	**Pres.**	**Rep.**
Gemma-4-E4B-it	non-thinking	1	64	0.95	0.05	0	1
	thinking	1	64	0.95	0.05	0	1
Qwen3.5-4B	non-thinking	0.7	20	0.8	0	1.5	1
	thinking	1	20	0.95	0	1.5	1
Kimi-VL-A3B-Thinking-2506	thinking	0.8	40	0.95	0.05	0	1
Kimi-VL-A3B-Instruct	non-thinking	0.2	40	0.95	0.05	0	1
Gemma-4-31B-it	non-thinking	1	64	0.95	0.05	0	1
	thinking	1	64	0.95	0.05	0	1
Qwen3.6-35B-A3B	non-thinking	0.7	20	0.8	0	1.5	1
	thinking	1	20	0.95	0	1.5	1
Muse-Glimmer-30B	thinking (high)	1	64	0.95	0.05	0	1
InternVL3.5-30B-A3B	non-thinking	0.8	40	0.95	0.05	0	1
	thinking	0.6	40	0.95	0.05	0	1
GLM-4.6V-Flash	non-thinking	0.8	2	0.6	0.05	0	1.1
	thinking	0.8	2	0.6	0.05	0	1.1

The decoding settings and modes are listed separately within each model group in Table 15. Muse-Glimmer-30B has only a thinking mode with different reasoning strengths (low, medium, high, xhigh) and we use the default (high) for decoding. Both the pilot set and main set share all decoding parameters in reference generation. In the columns, Temp. stands for temperature, Pres. for presence penalty and Rep. for repeat penalty. We use the default sampler without modification in llama.cpp.

K RUNTIME SETTINGS FOR FIDELITY METRICS COLLECTION

Table 16 lists the llama.cpp runtime of the teacher-forced evaluator that produced every fidelity record. The shared settings and their model- and run-specific exceptions are listed below.

Table 16: Runtime of the teacher-forced scorer (llama.cpp fork)

Setting	Value	Notes
Context length (`n_ctx`)	32,768	one sequence per run (`n_seq_max` 1), separate KV cache per sequence
Logical batch (`n_batch`)	2,048	
Micro-batch (`n_ubatch`)	512	2,048 for Gemma-4-31B-it
Teacher-forcing chunk (`tf_chunk`)	2,048	trajectory tokens decoded per teacher-forcing call
GPU offload (`n_gpu_layers`)	all layers	
Inference threads (`n_threads`)	8	
Metric threads (`metric_threads`)	8 or 12	does not affect the KLD computation or storage
Flash attention	on	
KV cache type (K / V)	f16 / f16	
SWA full cache (`swa_full`)	off	
Multi-token prediction	off	
Image token budget	native	`image_min_tokens` / `image_max_tokens` left at -1: the reference model's mtmd defaults
Media wrapper	stripped	
Prefix drift	not allowed	the replayed prefix must match the frozen trajectory token for token
Vocabulary attribute check	strict	waived (`allow_vocab_attr_mismatch`) for Gemma-4-31B-it and Gemma-4-E4B-it; the token mapping itself is still checked
Item order	by image count, ascending	
Scored answer prefix (`num_eval_tokens`)		2,048/8,192 for the pilot set; 1,024/2,048 for the main set
Build	commit masked for review; patch released with the artifacts	our fork of llama.cpp, CUDA 13.2, 76-byte metric record (VLMK v5)
Hardware	1 GPU per run	NVIDIA RTX PRO 6000 Blackwell Server Edition, driver 595.71.05

Every setting is echoed into the `collect_meta.json` of each scoring run together with the SHA-256 of the binary, its backend libraries and both GGUF files; the KV cache type is fixed by the campaign profile.

L POWER FUNCTION AND SAMPLE SIZE

Items, categories, and token caps. We use both the items and the pair categorization from the main set, pooling the six benchmarks within each mode. Non-thinking and thinking are analyzed separately. Close pairs comprise the tight and narrow categories; non-close pairs comprise the moderate and wide categories. Under cap rule defined in Section 5.1, close pairs use 512 tokens in non-thinking mode and 2,048 tokens in thinking mode, while non-close pairs use 256 and 1,024 tokens, respectively. All moments are evaluated at the selected cap; the primary results use main items, and the final table additionally reports pilot items as a sensitivity analysis. There are 635 non-thinking and 746 thinking candidate pairs. Since categorization and estimation use the same main items, the summaries describe these selected groups; they are not a held-out validation of a pilot design.

Paired and unpaired statistics. For a fixed candidate pair (θ_a, θ_b) and evaluation token cap T, we use the item-level divergences $D_i(\theta_a; T)$ and $D_i(\theta_b; T)$ from Eq. (4), and the paired differences $\Delta_i(T) = D_i(\theta_a; T) - D_i(\theta_b; T)$ from Eq. (5). As in Section 3.3, N is the number of retained paired items, $\widehat{\mu}_{ab}$ is their mean difference, and s_{ab} is their sample standard deviation. We suppress dependence on the fixed cap T below. The observed count N may differ across pairs. We use these paired observations to estimate the moments for both proposed sampling designs. Write s_a^2, s_b^2 for the sample variances of $D_i(\theta_a; T)$ and $D_i(\theta_b; T)$; these and s_{ab}^2 use denominator $N - 1$. Consistent with Eqs. (7) and (10), define

$$\begin{aligned} \widehat{\mathrm{SNR}}_{ab} &= \frac{\widehat{\mu}_{ab}}{s_{ab}}, & \widehat{\mathrm{SE}}_{ab} &= \frac{s_{ab}}{\sqrt{N}}, \\ \widehat{\mathrm{SE}}_{U,ab} &= \sqrt{\frac{s_a^2 + s_b^2}{N}}, & g_{ab} &= \frac{\widehat{\mathrm{SE}}_{U,ab}}{\widehat{\mathrm{SE}}_{ab}}. \end{aligned} \tag{12}$$

The unpaired design uses independently sampled items for the two candidates. Its estimated signed SNR is $\widehat{\mathrm{SNR}}_{U,ab} = \widehat{\mu}_{ab}/\sqrt{s_a^2 + s_b^2}$, computed directly from the mean difference and marginal variances. The quantity $\widehat{\mathrm{SE}}_{U,ab}$ is the estimated standard error implied by that design at N observations per candidate; the observed score vectors remain paired. The paired-to-unpaired SNR ratio from Appendix B is $g_{ab} = |\widehat{\mathrm{SNR}}_{ab}|/|\widehat{\mathrm{SNR}}_{U,ab}|$ when the effect is nonzero and both scales are positive. The standard-error ratio defines g_{ab} also at zero effect when the variances are positive. Table 17 reports the lower median of the per-pair ratios, with equal weight for each candidate pair.

Paired power and sample size. For independent, identically distributed normal differences, the population standardized gap is $\mathrm{SNR}_{ab} = \mu_{ab}/\sigma_{ab}$ as in Eq. (10). Following Section 3.4, let n denote a proposed item count per candidate. The paired-t statistic t_n on n common items follows

$$t_n \sim t_{n-1}(\lambda_n), \qquad \lambda_n = \sqrt{n}\,\mathrm{SNR}_{ab}. \tag{13}$$

Writing $F_{\nu,\lambda}$ for the noncentral-t CDF and $c_P = t_{1-\alpha/2,n-1}$, the two-sided power function used in Section 3.4 is

$$\pi_t(n, \mathrm{SNR}_{ab}; \alpha) = F_{n-1,\lambda_n}(-c_P) + 1 - F_{n-1,\lambda_n}(c_P). \tag{14}$$

This two-sided power is unchanged by reversing the sign of SNR_{ab}. Substituting $|\widehat{\mathrm{SNR}}_{ab}|$ gives the plug-in power $\pi_t(n, |\widehat{\mathrm{SNR}}_{ab}|; \alpha)$ used in Eq. (11). We use $\alpha = 0.05$ and target power 0.8. The implementation evaluates the two tails with SciPy's noncentral-t CDF and survival function. Non-finite tail evaluations fall back to the equivalent noncentral-F survival function, since $t_n^2 \sim F_{1,n-1}(\lambda_n^2)$. The paired calculation is exact under this normal-difference model at the supplied inputs; inserting estimated moments does not guarantee 80 % power for future data.

Power for independent item samples. At a proposed sample size n, the unpaired design draws n independent, identically distributed items for each candidate, with the two samples independent of one another. Both samples use the same benchmark mixture and the selected token cap. Using the marginal variances estimated from the observed paired items, define the Welch–Satterthwaite degrees of freedom by

$$\nu_W(n) = (n-1)\frac{(s_a^2 + s_b^2)^2}{(s_a^2)^2 + (s_b^2)^2}. \tag{15}$$

For two-sided power, we use a noncentral t approximation with degrees of freedom $\nu_W(n)$ and noncentrality magnitude $\lambda_{U,n} = \sqrt{n}\,|\widehat{\mathrm{SNR}}_{U,ab}|$. Writing $c_U = t_{1-\alpha/2,\nu_W(n)}$, the implemented working power is

$$\pi_U(n) = F_{\nu_W(n),\lambda_{U,n}}(-c_U) + 1 - F_{\nu_W(n),\lambda_{U,n}}(c_U). \tag{16}$$

This calculation holds the estimated mean difference and marginal variances fixed as n varies. It is a plug-in Welch–Satterthwaite approximation: the sample variances and degrees of freedom of an actual Welch test are random, so this is not its exact power law even for normal scores. Estimating the inputs from paired observations does not require the proposed future samples to be paired, and the resulting estimate is not an empirical validation using independently collected samples.

Here n counts observations per candidate in both designs. The paired design uses n common items; the independent-sample design uses two independent samples of n items. Each design produces $2n$ candidate scores. This budget counts candidate scores; reference generation and item acquisition can have different costs across the designs.

Search limit and censored estimates. For compactness, write $\pi_P(n) = \pi_t(n, |\widehat{\mathrm{SNR}}_{ab}|; \alpha)$ for the paired plug-in power. For each analysis $A \in \{P, U\}$, the reported estimate searches for

$$\widehat{N}_{80,A} = \min\{n \in \mathbb{Z} : 2 \le n \le L_{ab},\ \pi_A(n) \ge 0.8\}, \qquad L_{ab} = 10N. \tag{17}$$

A doubling search followed by integer bisection is used. At each proposed n, the independent-sample calculation recomputes both $\lambda_{U,n}$ and $\nu_W(n)$ in Eq. (16). The paired estimate $\widehat{N}_{80,P}$ is the search-limited version of $\widehat{N}_{80}$ in Eq. (11). The limit L_{ab} restricts extrapolation to ten times that pair's observed item count. A crossing within the limit is *finite*; no crossing by the limit is *censored*; zero estimated effect within a defined working model is *unattainable*; an undefined working model is *undefined*. Unattainable does not establish that the population effect is zero.

A constant nonzero paired difference uses the limiting value $\widehat{N}_{80,P} = 2$. The independent-sample calculation remains defined in this case whenever $s_a^2 + s_b^2 > 0$, because it uses $|\widehat{\mathrm{SNR}}_{U,ab}|$ directly and does not divide the paired SNR by the SNR ratio. If exactly one marginal variance is zero, the independent-sample model is defined with $\nu_W(n) = n - 1$. If both marginal variances are zero, its SNR or degrees of freedom is undefined; we report *undefined* without extending the Welch model to this deterministic case. Non-finite required moments also give an undefined working model.

For group medians, a finite estimate contributes $[\widehat{N}_{80,A}, \widehat{N}_{80,A}]$, a censored estimate contributes $[L_{ab} + 1, \infty]$, and an unattainable estimate contributes $[\infty, \infty]$. Undefined models are excluded and counted. The lower medians of the lower and upper endpoints bound the group median. Coincident finite bounds identify a finite median, unequal bounds give censored, and two infinite bounds give unattainable. With no eligible pair the median is undefined. Thus a censored estimate is not assigned a known infinite sample size. For k values, the lower median is the order statistic at zero-based index $\lfloor (k-1)/2 \rfloor$.

Bootstrap intervals. The brackets in Table 17 are nominal 95 % percentile intervals from 5,000 joint item bootstrap resamples (seed 0). Within each mode, each replicate resamples the pooled union of main items and applies the same item weights to all models, candidates, and caps, retaining each model's available-item mask. Each replicate recomputes the mean difference, marginal and difference variances, both SNRs, the Welch variance ratio, and both sample-size estimates. Pair membership, main-set categories, caps, and the original limits L_{ab} remain fixed. Pairs and models are not resampled independently. The resamples describe uncertainty in moments estimated from the observed paired data, so they preserve joint item dependence even though the unpaired power calculation describes a future independent-sample design.

The lower median is recomputed in each replicate while retaining censoring bounds. Each analysis uses its own roster of pairs with an originally defined working model. At 5,000 replicates, the percentile endpoints use the 125th and 4,875th ordered values. For each percentile endpoint, equal finite bounds identify a finite value, unequal bounds give censored, and two infinite bounds give unattainable; no eligible pair gives undefined. If an originally eligible pair becomes undefined in any replicate, that group's interval is undefined; invalid replicates are not discarded. The retained table has no invalid replicates. These intervals condition on the selected categories and do not establish exact 95 % coverage.

Table 17: Reproduction of Table 1 for reference alongside the definitions of this appendix. Items and pair categories come from the main set, pooled across six benchmarks within each mode, at the caps of cap rule defined in Section 5.1. Brackets give nominal 95% joint item-bootstrap intervals (5,000 resamples, seed 0).

			Pooled benchmarks		
Mode	Category	Pairs	SNR ratio	$\widehat{N}_{80}$ paired	$\widehat{N}_{80}$ independent
Non-thinking	close	81	2.01	10949 [5155, 13117]	censored [censored, censored]
Non-thinking	non-close	554	1.41	149 [120, 162]	318 [273, 352]
Thinking	close	129	3.68	353 [301, 451]	5471 [4869, 6175]
Thinking	non-close	617	2.02	7 [6, 7]	18 [16, 19]

The table's paired-to-unpaired SNR ratio medians are larger for thinking than for non-thinking and for close than for non-close pairs. Median paired sample sizes range from 7 to 10,949 items. For thinking close pairs, the ratio of the unpaired median to the paired median is $5471/353 \approx 15.5$; this is a ratio of group medians, not a median of per-pair sample-size ratios. For non-thinking close pairs, 51 of the 81 unpaired estimates are censored and 30 are finite. Their group median has identification bounds $[22041, \infty]$ under the pair-specific limits. Both bootstrap percentile endpoints are censored, so the interval is reported as $[\text{censored}, \text{censored}]$. There is no single common $10N$ threshold for all pairs.

Large-sample approximation. For paired power, the usual normal approximation gives

$$\widehat{N}_{80,P} \approx \left\lceil \frac{(z_{1-\alpha/2} + z_{0.8})^2}{|\widehat{\mathrm{SNR}}_{ab}|^2} \right\rceil. \tag{18}$$

Here z_q is the q-quantile of the standard normal distribution. At $|\widehat{\mathrm{SNR}}_{ab}| = 2, 1, 0.5, 0.2$, the exact paired calculations give 5, 10, 34, and 199 items, compared with 2, 8, 32, and 197 from this approximation. We use the approximation only to illustrate inverse-square dependence on SNR. The independent-sample analogue replaces $|\widehat{\mathrm{SNR}}_{ab}|$ with $|\widehat{\mathrm{SNR}}_{U,ab}|$. When both SNRs are finite and nonzero, the resulting large-sample ratio is $\widehat{N}_{80,U}/\widehat{N}_{80,P} \approx g_{ab}^2$ before integer rounding. The table's paired estimates use Eq. (14), and its unpaired estimates use Eq. (16); neither the squared SNR ratio nor the normal approximation is substituted for the independent-sample Welch search.

Paired relative minimum detectable effect. For each proposed budget n, we invert the paired power function. Let $\mathrm{SNR}_n^* > 0$ satisfy $\pi_t(n, \mathrm{SNR}_n^*; 0.05) = 0.8$. Using the corpus means from Eq. (4), define

$$R_{ab} = \frac{s_{ab}}{\frac{1}{2}\big(\overline{D}(\theta_a; T) + \overline{D}(\theta_b; T)\big)}, \qquad \widehat{\mathrm{MDE}}_{ab}(n) = \mathrm{SNR}_n^* R_{ab}. \tag{19}$$

Here R_{ab} is the noise-to-divergence ratio; it differs from the correlation r_{ab} and the paired-to-unpaired SNR ratio g_{ab} in Appendix B. For the primary MDE table, all moments are estimated on main items at the selected cap T under the cap rule in Section 5.1. The denominator is the same pair-mean divergence used to estimate $\rho_{ab}(T)$ in Eq. (9), so $\widehat{\mathrm{MDE}}_{ab}(n)$ is expressed on the scale of $|\rho_{ab}(T)|$. This is the relative absolute mean gap detected with 80 % power by the two-sided paired test under its normal-difference model with the estimated scale held fixed. It concerns detecting a nonzero difference in either direction, not equivalence or non-inferiority. These MDEs do not describe the unpaired test and do not guarantee empirical power under heavy tails or distribution changes.

Table 18 reports lower medians across all pairs in each mode. Table 19 compares main and pilot items while keeping the main-set categories and their selected caps T fixed. Each pair has equal weight; pairs sharing items and candidates are not independent replicates. All quantiles use the lower order statistic without interpolation. Percentages use one decimal place below 10 % and whole percentages otherwise.

Table 18: Paired relative minimum detectable effects estimated on the main set, with main-set categories and cap rule defined in Section 5.1. Entries are lower medians across all candidate pairs in each mode, at 80 % model power and two-sided $\alpha = 0.05$.

	Items per pair				
Mode	100	250	500	1,000	3,000
Non-thinking	34 %	22 %	15 %	11 %	6.2 %
Thinking	7.8 %	4.9 %	3.5 %	2.4 %	1.4 %

Table 19: Sensitivity of paired MDEs to the item set, with main-set categories and cap rule defined in Section 5.1 fixed. Each item set contains the same 635 non-thinking and 746 thinking candidate pairs. Panel (a) reports lower median relative MDEs at 80 % model power and two-sided $\alpha = 0.05$. Panel (b) reports lower median item counts and lower-quantile summaries of R_{ab} from Eq. (19). Main is the primary analysis; pilot is an additional sensitivity analysis.

(a) Paired relative minimum detectable effect

		Items per pair				
Mode	Item set	100	250	500	1,000	3,000
Non-thinking	Main	34 %	22 %	15 %	11 %	6.2 %
Non-thinking	Pilot	35 %	22 %	15 %	11 %	6.3 %
Thinking	Main	7.8 %	4.9 %	3.5 %	2.4 %	1.4 %
Thinking	Pilot	6.8 %	4.3 %	3.0 %	2.1 %	1.2 %

(b) Items and noise-to-divergence ratio

			R_{ab}	
Mode	Item set	Items	Median	IQR
Non-thinking	Main	2,376	1.22	0.93–2.11
Non-thinking	Pilot	593	1.23	0.90–1.84
Thinking	Main	2,375	0.28	0.20–0.38
Thinking	Pilot	592	0.24	0.19–0.36

Sensitivity to the item set. The primary estimates use main items. As an additional sensitivity analysis, Table 19 recomputes paired MDEs and noise-to-divergence ratios on pilot items for the same candidate pairs, with the main-set categories and caps held fixed. Pilot and main observations are analyzed separately and never pooled. This comparison measures sensitivity to the item set; it is not a prospective pilot plan or an independent validation of category selection, because the categories were selected on the main set. The MDE estimates remain conditional diagnostics under the paired power model.

M FULL GGUF CHECKPOINTS

Table 20: GGUF checkpoints used to collect fidelity metrics

Quantization	GGUF filename
Gemma-4-E4B-it	
Reference	
Source: bartowski/google_gemma-4-E4B-it-GGUF (029e94146).	
mmproj:	mmproj-google_gemma-4-E4B-it-bf16.gguf
BF16	google_gemma-4-E4B-it-bf16.gguf
Candidates	
Source: bartowski/google_gemma-4-E4B-it-GGUF (029e94146).	
Shared mmproj:	mmproj-google_gemma-4-E4B-it-bf16.gguf
IQ4_NL	google_gemma-4-E4B-it-IQ4_NL.gguf
IQ4_XS	google_gemma-4-E4B-it-IQ4_XS.gguf
Q3_K_XL	google_gemma-4-E4B-it-Q3_K_XL.gguf
Q4_0	google_gemma-4-E4B-it-Q4_0.gguf
Q4_1	google_gemma-4-E4B-it-Q4_1.gguf
Q4_K_M	google_gemma-4-E4B-it-Q4_K_M.gguf
Q4_K_S	google_gemma-4-E4B-it-Q4_K_S.gguf
Source: unsloth/gemma-4-E4B-it-GGUF (bfc15c382).	
Shared mmproj:	mmproj-BF16.gguf
IQ4_NL	gemma-4-E4B-it-IQ4_NL.gguf
IQ4_XS	gemma-4-E4B-it-IQ4_XS.gguf
Q4_0	gemma-4-E4B-it-Q4_0.gguf
Q4_1	gemma-4-E4B-it-Q4_1.gguf
Q4_K_M	gemma-4-E4B-it-Q4_K_M.gguf
Q4_K_S	gemma-4-E4B-it-Q4_K_S.gguf
UD-Q3_K_XL	gemma-4-E4B-it-UD-Q3_K_XL.gguf
UD-Q4_K_XL	gemma-4-E4B-it-UD-Q4_K_XL.gguf
Source: google/gemma-4-E4B-it-qat-q4_0-gguf (4b4a2c1d5).	
Shared mmproj:	gemma-4-E4B-it-mmproj.gguf
QAT-Q4_0	gemma-4-E4B_q4_0-it.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
Qwen3.5-4B	
Reference	
Source: bartowski/Qwen_Qwen3.5-4B-GGUF (4168f45a1).	
mmproj:	mmproj-Qwen_Qwen3.5-4B-bf16.gguf
BF16	Qwen_Qwen3.5-4B-bf16.gguf
Candidates	
Source: bartowski/Qwen_Qwen3.5-4B-GGUF (4168f45a1).	
Shared mmproj:	mmproj-Qwen_Qwen3.5-4B-bf16.gguf
IQ4_NL	Qwen_Qwen3.5-4B-IQ4_NL.gguf
IQ4_XS	Qwen_Qwen3.5-4B-IQ4_XS.gguf
Q4_0	Qwen_Qwen3.5-4B-Q4_0.gguf
Q4_1	Qwen_Qwen3.5-4B-Q4_1.gguf
Q4_K_M	Qwen_Qwen3.5-4B-Q4_K_M.gguf
Q4_K_S	Qwen_Qwen3.5-4B-Q4_K_S.gguf
Source: unsloth/Qwen3.5-4B-GGUF (e87f17647).	
Shared mmproj:	mmproj-BF16.gguf
IQ4_NL	Qwen3.5-4B-IQ4_NL.gguf
IQ4_XS	Qwen3.5-4B-IQ4_XS.gguf
Q4_0	Qwen3.5-4B-Q4_0.gguf
Q4_1	Qwen3.5-4B-Q4_1.gguf
Q4_K_M	Qwen3.5-4B-Q4_K_M.gguf
Q4_K_S	Qwen3.5-4B-Q4_K_S.gguf
UD-Q4_K_XL	Qwen3.5-4B-UD-Q4_K_XL.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
Kimi-VL-A3B-Thinking-2506	
Reference	
Source: Locally produced (llama.cpp commit: 6a1a922d2).	
mmproj:	mmproj-Kimi-VL-A3B-Thinking-2506-bf16.gguf
BF16	Kimi-VL-A3B-Thinking-2506-bf16.gguf
Candidates	
Source: Locally produced (llama.cpp commit: 6a1a922d2).	
Shared mmproj:	mmproj-Kimi-VL-A3B-Thinking-2506-bf16.gguf
Q4_0	Kimi-VL-A3B-Thinking-2506-Q4_0.gguf
Q4_1	Kimi-VL-A3B-Thinking-2506-Q4_1.gguf
Q4_K_M	Kimi-VL-A3B-Thinking-2506-Q4_K_M.gguf
Q4_K_S	Kimi-VL-A3B-Thinking-2506-Q4_K_S.gguf
Source: mradermacher/Kimi-VL-A3B-Thinking-2506-GGUF (abcc7e983).	
Source: mradermacher/Kimi-VL-A3B-Thinking-2506-i1-GGUF (c896f2583).	
Shared mmproj:	Kimi-VL-A3B-Thinking-2506.mmproj-f16.gguf
Q4_K_M	Kimi-VL-A3B-Thinking-2506.Q4_K_M.gguf
Q4_K_S	Kimi-VL-A3B-Thinking-2506.Q4_K_S.gguf
i1-IQ4_NL	Kimi-VL-A3B-Thinking-2506.i1-IQ4_NL.gguf
i1-IQ4_XS	Kimi-VL-A3B-Thinking-2506.i1-IQ4_XS.gguf
i1-Q4_0	Kimi-VL-A3B-Thinking-2506.i1-Q4_0.gguf
i1-Q4_1	Kimi-VL-A3B-Thinking-2506.i1-Q4_1.gguf
i1-Q4_K_M	Kimi-VL-A3B-Thinking-2506.i1-Q4_K_M.gguf
i1-Q4_K_S	Kimi-VL-A3B-Thinking-2506.i1-Q4_K_S.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
Kimi-VL-A3B-Instruct	
Reference	
Source: Locally produced (llama.cpp commit: 6a1a922d2).	
mmproj:	mmproj-Kimi-VL-A3B-Instruct-bf16.gguf
BF16	Kimi-VL-A3B-Instruct-bf16.gguf
Candidates	
Source: Locally produced (llama.cpp commit: 6a1a922d2).	
Shared mmproj:	mmproj-Kimi-VL-A3B-Instruct-bf16.gguf
IQ4_NL	Kimi-VL-A3B-Instruct-IQ4_NL.gguf
IQ4_XS	Kimi-VL-A3B-Instruct-IQ4_XS.gguf
Q4_0	Kimi-VL-A3B-Instruct-Q4_0.gguf
Q4_1	Kimi-VL-A3B-Instruct-Q4_1.gguf
Q4_K_M	Kimi-VL-A3B-Instruct-Q4_K_M.gguf
Q4_K_S	Kimi-VL-A3B-Instruct-Q4_K_S.gguf
Source: mradermacher/Kimi-VL-A3B-Instruct-GGUF (d645665be).	
Source: mradermacher/Kimi-VL-A3B-Instruct-i1-GGUF (e54dd1379).	
Shared mmproj:	Kimi-VL-A3B-Instruct.mmproj-f16.gguf
IQ4_XS	Kimi-VL-A3B-Instruct.IQ4_XS.gguf
Q4_K_M	Kimi-VL-A3B-Instruct.Q4_K_M.gguf
Q4_K_S	Kimi-VL-A3B-Instruct.Q4_K_S.gguf
i1-IQ4_NL	Kimi-VL-A3B-Instruct.i1-IQ4_NL.gguf
i1-IQ4_XS	Kimi-VL-A3B-Instruct.i1-IQ4_XS.gguf
i1-Q4_0	Kimi-VL-A3B-Instruct.i1-Q4_0.gguf
i1-Q4_1	Kimi-VL-A3B-Instruct.i1-Q4_1.gguf
i1-Q4_K_M	Kimi-VL-A3B-Instruct.i1-Q4_K_M.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
Gemma-4-31B-it	
Reference	
Source: bartowski/google_gemma-4-31B-it-GGUF (619aca758).	
mmproj:	mmproj-google_gemma-4-31B-it-bf16.gguf
BF16	google_gemma-4-31B-it-bf16-00001-of-00002.gguf
	google_gemma-4-31B-it-bf16-00002-of-00002.gguf
Candidates	
Source: bartowski/google_gemma-4-31B-it-GGUF (619aca758).	
Shared mmproj:	mmproj-google_gemma-4-31B-it-bf16.gguf
IQ4_XS	google_gemma-4-31B-it-IQ4_XS.gguf
Q3_K_XL	google_gemma-4-31B-it-Q3_K_XL.gguf
Q4_0	google_gemma-4-31B-it-Q4_0.gguf
Q4_1	google_gemma-4-31B-it-Q4_1.gguf
Q4_K_M	google_gemma-4-31B-it-Q4_K_M.gguf
Q4_K_S	google_gemma-4-31B-it-Q4_K_S.gguf
Source: unsloth/gemma-4-31B-it-GGUF (c1ac76e99).	
Shared mmproj:	mmproj-BF16.gguf
IQ4_NL	gemma-4-31B-it-IQ4_NL.gguf
IQ4_XS	gemma-4-31B-it-IQ4_XS.gguf
Q4_0	gemma-4-31B-it-Q4_0.gguf
Q4_1	gemma-4-31B-it-Q4_1.gguf
Q4_K_M	gemma-4-31B-it-Q4_K_M.gguf
Q4_K_S	gemma-4-31B-it-Q4_K_S.gguf
UD-Q3_K_XL	gemma-4-31B-it-UD-Q3_K_XL.gguf
UD-Q4_K_XL	gemma-4-31B-it-UD-Q4_K_XL.gguf
Source: google/gemma-4-31B-it-qat-q4_0-gguf (59dde2457).	
Shared mmproj:	gemma-4-31B-it-mmproj.gguf
QAT-Q4_0	gemma-4-31B_q4_0-it.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
Qwen3.6-35B-A3B	
Reference	
Source: bartowski/Qwen_Qwen3.6-35B-A3B-GGUF (5c2410d71).	
mmproj:	mmproj-Qwen_Qwen3.6-35B-A3B-bf16.gguf
BF16	Qwen_Qwen3.6-35B-A3B-bf16-00001-of-00002.gguf
	Qwen_Qwen3.6-35B-A3B-bf16-00002-of-00002.gguf
Candidates	
Source: bartowski/Qwen_Qwen3.6-35B-A3B-GGUF (5c2410d71).	
Shared mmproj:	mmproj-Qwen_Qwen3.6-35B-A3B-bf16.gguf
IQ4_NL	Qwen_Qwen3.6-35B-A3B-IQ4_NL.gguf
IQ4_XS	Qwen_Qwen3.6-35B-A3B-IQ4_XS.gguf
Q4_0	Qwen_Qwen3.6-35B-A3B-Q4_0.gguf
Q4_1	Qwen_Qwen3.6-35B-A3B-Q4_1.gguf
Q4_K_L	Qwen_Qwen3.6-35B-A3B-Q4_K_L.gguf
Q4_K_M	Qwen_Qwen3.6-35B-A3B-Q4_K_M.gguf
Q4_K_S	Qwen_Qwen3.6-35B-A3B-Q4_K_S.gguf
Source: unsloth/Qwen3.6-35B-A3B-GGUF (a483e9e6c).	
Shared mmproj:	mmproj-BF16.gguf
UD-Q4_K_M	Qwen3.6-35B-A3B-UD-Q4_K_M.gguf
UD-Q4_K_S	Qwen3.6-35B-A3B-UD-Q4_K_S.gguf
UD-Q4_K_XL	Qwen3.6-35B-A3B-UD-Q4_K_XL.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
InternVL3.5-30B-A3B	
Reference	
Source: bartowski/OpenGVLab_InternVL3_5-30B-A3B-GGUF (a6824f88f).	
mmproj:	mmproj-OpenGVLab_InternVL3_5-30B-A3B-bf16.gguf
BF16	OpenGVLab_InternVL3_5-30B-A3B-bf16-00001-of-00002.gguf
	OpenGVLab_InternVL3_5-30B-A3B-bf16-00002-of-00002.gguf
Candidates	
Source: bartowski/OpenGVLab_InternVL3_5-30B-A3B-GGUF (a6824f88f).	
Shared mmproj:	mmproj-OpenGVLab_InternVL3_5-30B-A3B-bf16.gguf
IQ4_NL	OpenGVLab_InternVL3_5-30B-A3B-IQ4_NL.gguf
IQ4_XS	OpenGVLab_InternVL3_5-30B-A3B-IQ4_XS.gguf
Q3_K_XL	OpenGVLab_InternVL3_5-30B-A3B-Q3_K_XL.gguf
Q4_0	OpenGVLab_InternVL3_5-30B-A3B-Q4_0.gguf
Q4_1	OpenGVLab_InternVL3_5-30B-A3B-Q4_1.gguf
Q4_K_M	OpenGVLab_InternVL3_5-30B-A3B-Q4_K_M.gguf
Q4_K_S	OpenGVLab_InternVL3_5-30B-A3B-Q4_K_S.gguf
Source: mradermacher/InternVL3_5-30B-A3B-GGUF (21b0a0cae).	
Source: mradermacher/InternVL3_5-30B-A3B-i1-GGUF (120d600e4).	
Shared mmproj:	InternVL3_5-30B-A3B.mmproj-f16.gguf
IQ4_XS	InternVL3_5-30B-A3B.IQ4_XS.gguf
Q4_K_M	InternVL3_5-30B-A3B.Q4_K_M.gguf
Q4_K_S	InternVL3_5-30B-A3B.Q4_K_S.gguf
i1-IQ4_XS	InternVL3_5-30B-A3B.i1-IQ4_XS.gguf
i1-Q4_0	InternVL3_5-30B-A3B.i1-Q4_0.gguf
i1-Q4_1	InternVL3_5-30B-A3B.i1-Q4_1.gguf
i1-Q4_K_M	InternVL3_5-30B-A3B.i1-Q4_K_M.gguf
i1-Q4_K_S	InternVL3_5-30B-A3B.i1-Q4_K_S.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
GLM-4.6V-Flash	
Reference	
Source: bartowski/zai-org_GLM-4.6V-Flash-GGUF (6a3b3ca7b).	
mmproj:	mmproj-zai-org_GLM-4.6V-Flash-bf16.gguf
BF16	zai-org_GLM-4.6V-Flash-bf16.gguf
Candidates	
Source: bartowski/zai-org_GLM-4.6V-Flash-GGUF (6a3b3ca7b).	
Shared mmproj:	mmproj-zai-org_GLM-4.6V-Flash-bf16.gguf
IQ4_NL	zai-org_GLM-4.6V-Flash-IQ4_NL.gguf
IQ4_XS	zai-org_GLM-4.6V-Flash-IQ4_XS.gguf
Q3_K_XL	zai-org_GLM-4.6V-Flash-Q3_K_XL.gguf
Q4_0	zai-org_GLM-4.6V-Flash-Q4_0.gguf
Q4_1	zai-org_GLM-4.6V-Flash-Q4_1.gguf
Q4_K_M	zai-org_GLM-4.6V-Flash-Q4_K_M.gguf
Q4_K_S	zai-org_GLM-4.6V-Flash-Q4_K_S.gguf
Source: unsloth/GLM-4.6V-Flash-GGUF (c78a0727c).	
Shared mmproj:	mmproj-BF16.gguf
IQ4_NL	GLM-4.6V-Flash-IQ4_NL.gguf
IQ4_XS	GLM-4.6V-Flash-IQ4_XS.gguf
Q4_0	GLM-4.6V-Flash-Q4_0.gguf
Q4_1	GLM-4.6V-Flash-Q4_1.gguf
Q4_K_M	GLM-4.6V-Flash-Q4_K_M.gguf
Q4_K_S	GLM-4.6V-Flash-Q4_K_S.gguf
UD-Q4_K_XL	GLM-4.6V-Flash-UD-Q4_K_XL.gguf

Table 20: GGUF checkpoints used to collect fidelity metrics (continued)

Quantization	GGUF filename
Muse-Glimmer-30B	
Reference	
Source: bartowski/Muse-Glimmer-30B-GGUF (01e1b7c30).	
mmproj:	mmproj-Muse-Glimmer-30B-bf16.gguf
BF16	Muse-Glimmer-30B-bf16-00001-of-00002.gguf
	Muse-Glimmer-30B-bf16-00002-of-00002.gguf
Candidates	
Source: bartowski/Muse-Glimmer-30B-GGUF (01e1b7c30).	
Shared mmproj:	mmproj-Muse-Glimmer-30B-bf16.gguf
IQ4_NL	Muse-Glimmer-30B-IQ4_NL.gguf
IQ4_XS	Muse-Glimmer-30B-IQ4_XS.gguf
Q4_0	Muse-Glimmer-30B-Q4_0.gguf
Q4_1	Muse-Glimmer-30B-Q4_1.gguf
Q4_K_M	Muse-Glimmer-30B-Q4_K_M.gguf
Q4_K_S	Muse-Glimmer-30B-Q4_K_S.gguf
Source: unsloth/Muse-Glimmer-30B-GGUF (faa5b025c).	
Shared mmproj:	mmproj-Muse-Glimmer-30B-BF16.gguf
UD-Q3_K_XL	Muse-Glimmer-30B-UD-Q3_K_XL.gguf
UD-Q4_K_XL	Muse-Glimmer-30B-UD-Q4_K_XL.gguf
Source: mradermacher/Muse-Glimmer-30B-GGUF (265f0f620).	
Source: mradermacher/Muse-Glimmer-30B-i1-GGUF (c256bff31).	
Shared mmproj:	Muse-Glimmer-30B.mmproj-f16.gguf
IQ4_XS	Muse-Glimmer-30B.IQ4_XS.gguf
Q3_K_L	Muse-Glimmer-30B.Q3_K_L.gguf
Q4_K_M	Muse-Glimmer-30B.Q4_K_M.gguf
Q4_K_S	Muse-Glimmer-30B.Q4_K_S.gguf
i1-IQ4_XS	Muse-Glimmer-30B.i1-IQ4_XS.gguf
i1-Q4_0	Muse-Glimmer-30B.i1-Q4_0.gguf
i1-Q4_1	Muse-Glimmer-30B.i1-Q4_1.gguf
i1-Q4_K_M	Muse-Glimmer-30B.i1-Q4_K_M.gguf
i1-Q4_K_S	Muse-Glimmer-30B.i1-Q4_K_S.gguf