What Users Leave Unsaid: Under-Specified Queries Limit Vision-Language Models

Dasol Choi[1,2*] **Guijin Son**[3*] **Hanwool Lee**[1*] **Minhyuk Kim**[4] **Hyunwoo Ko**[3]
Teabin Lim[5] **Eungyeol Ahn**[5] **Jungwhan Kim**[6] **Seunghyeok Hong**[7†] **Youngsook Song**[8†]

[1]AIM Intelligence [2]Yonsei University [3]OneLineAI [4]Korea University
[5]Doodlin Corp. [6]NAVER Cloud [7]Hankuk University of Foreign Studies [8]Lablup Inc.

GitHub HuggingFace Leaderboard

dasolchoi@yonsei.ac.kr, spthsrbwls123@yonsei.ac.kr

Abstract

Current vision-language benchmarks predominantly feature well-structured questions with clear, explicit prompts. However, real user queries are often informal and underspecified. Users naturally leave much unsaid, relying on images to convey context. We introduce HAERAE-Vision, a benchmark of 653 real-world visual questions from Korean online communities (0.76% survival from 86K candidates), each paired with an explicit rewrite, yielding 1,306 query variants in total. Evaluating 45 VLMs, we find that even state-of-the-art models (GPT-5, Gemini 2.5 Pro) achieve under 50% on the original queries. Crucially, query explicitation alone yields 8 to 22 point improvements, with smaller models benefiting most. We further show that even with web search, under-specified queries underperform explicit queries without search, revealing that current retrieval cannot compensate for what users leave unsaid. Our findings demonstrate that a substantial portion of VLM difficulty stems from natural query under-specification instead of model capability, highlighting a critical gap between benchmark evaluation and real-world deployment.

1 Introduction

When users ask visual questions, they rarely provide complete, well-structured queries. Instead, they write informally, omit context, and rely on images to convey what they leave unsaid. A user might ask "How do I do this?" alongside an image, expecting the responder to identify the problem, infer the relevant domain, and provide a step-by-step solution. This natural tendency toward under-specification poses a fundamental challenge for vision-language models (VLMs) (Li et al., 2025), yet current benchmarks predominantly feature clean, explicit prompts failing to capture this phenomenon (Kim and Jung, 2025; Ju et al., 2024).

We introduce HAERAE-Vision, a benchmark constructed from authentic user queries in Korean online communities. Starting from 86,052 question-image pairs across nine platforms, we apply a six-stage filtering pipeline to yield 653 rigorously validated items (0.76% survival rate). The resulting questions are ambiguous, informal, and under-specified, mirroring the noisy nature of authentic multimodal interactions. To isolate the effect of query under-specification, we additionally construct HAERAE-Vision-Explicit, a parallel dataset where each question is systematically rewritten to state the missing information explicitly.

Our experiments reveal that query explicitation alone yields up to 22 point improvements across models, with smaller models benefiting most dramatically. Even state-of-the-art models achieve under 50% on original queries but surpass 55% with explicitation (GPT-5: 48.0%→57.6%, Gemini 2.5 Pro: 48.5%→56.7%). Furthermore, we demonstrate that even with web search enabled, under-specified queries still underperform explicit queries without search. This reveals that current retrieval systems cannot compensate for what users leave unsaid, as models must first understand user intent before search becomes effective.

These findings challenge a common assumption in VLM evaluation: that benchmark difficulty reflects model capability limitations. We show that a substantial portion of difficulty stems instead from the natural under-specification of user queries, highlighting a critical gap between benchmark evaluation and real-world deployment.

Our contributions are:

- **Real-world query benchmark:** HAERAE-Vision, comprising 653 user-generated visual questions, filtered from 86K candidates (0.76% survival), spanning 13 domains.
- **Paired explicit rewrites:** A parallel dataset

[*]Equal contribution.
[†]Corresponding authors.

Findings of the Association for Computational Linguistics: ACL 2026, pages 10863–10886

Figure 1: **Representative examples from HAERAE-Vision across six of the 13 domains.** Each example shows an under-specified Korean question with English translation, the corresponding image, and evaluation checklist criteria. Note the informal, context-dependent nature of the original queries.

of clarified queries enabling controlled measurement of under-specification effects.

- **Quantifying under-specification:** Empirical evidence that explicitation yields up to 22% improvements, with smaller models benefiting most. This demonstrates that query ambiguity accounts for substantial VLM difficulty.

2 HAERAE-Vision Benchmark

We present HAERAE-Vision, a benchmark constructed from authentic user queries, designed to capture the under-specified, informal nature of real-world visual questions. Our six-stage pipeline transforms large-scale, noisy community data into high-quality evaluation problems while preserving the natural characteristics of user queries.

2.1 Dataset Construction Pipeline

Starting from 86,052 raw question-image pairs from nine Korean platforms spanning general Q&A, gaming, science, and coding forums (see Appendix A.1 for detailed platform descriptions), we obtain 653 high-quality problems (0.76% survival rate). Figure 2 illustrates the filtering process.

Stage 1: Data Collection. We collect (question, image, answer) triplets, prioritizing those with an accepted answer rewarded by the asker or with high online engagement (views, likes, comments), targeting questions the community finds valuable.

Stage 2: Appropriateness Filtering. Each triplet is screened along three axes: (i) content safety (political/religious material, discrimination, adult content), (ii) objectivity (overly subjective or unverifiable prompts), and (iii) time-sensitiveness. GPT-4o is used for the automated filtering, flagging problematic items while allowing borderline cases to proceed to human validation. This removes 49.6% of raw data (see Appendix B.1).

Stage 3: Difficulty Calibration. Following prior benchmarks (Zellers et al., 2019; Hendrycks et al., 2021), we remove questions that strong models solve trivially. Three models (GPT-4o, Gemini-1.5-Flash, Claude-3.5) are evaluated against community-provided human answers using semantic-overlap scoring. Items with an average score above 0.6 are removed, eliminating 87.6% of the remaining items.

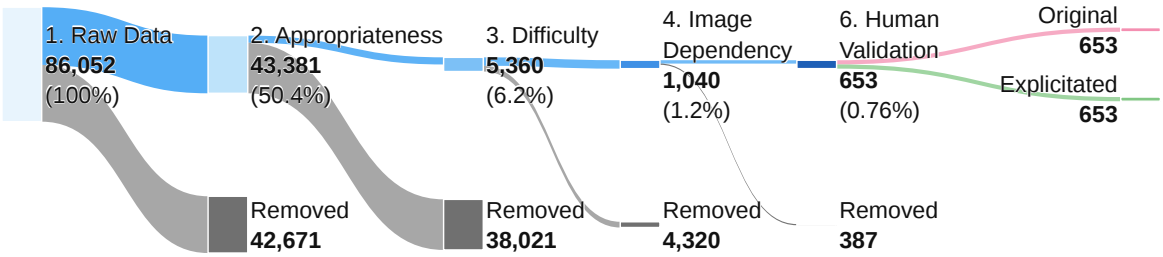


Figure 2: **Filtering pipeline showing data reduction at each stage.** Numbers indicate pipeline stages described in Section 2.1. The 0.76% survival rate reflects rigorous quality control. Each validated question is paired with an explicitated rewrite, yielding 1,306 query variants.

Stage 4: Image Dependency Verification. To confirm that each question requires visual reasoning, we generate two responses per item using Gemini 2.0 Flash: one with the image and one without. Both responses are evaluated against the human reference, and items where the quality gap is below 1 point (on a 0-10 scale) are discarded as image-independent (see Appendix B.2).

Stage 5: Checklist Generation. Each answer is converted into a structured checklist with 1 to 5 criteria using o4-mini. The model is instructed to define the minimal necessary elements for a response to be deemed correct, focusing on correctness, explanation quality, and reasoning steps rather than exhaustive coverage. This design enables partial-credit scoring and ensures reproducible, automated evaluation across models (see Appendix B.3).

Stage 6: Human Validation. Seven native Korean annotators conduct three-phase validation: (1) filtering based on image appropriateness, question clarity, and checklist validity, removing any item flagged by at least one annotator; (2) refinement of questions and LLM-generated checklists, where annotators rewrite unclear criteria and remove items not grounded in the original question–image pair; (3) final audit for category consolidation and consistency. This removes 37.2% of remaining items, yielding 653 problems (see Appendix C.1).

2.2 Dataset Statistics

Our final benchmark contains 653 problems with an average of 3.3 checklist items and 1.3 images per question. Table 1 presents the distribution across 13 categories, where Natural Objects and Gaming are the most represented. The survival rate per platform varies significantly (0.2% to 14.4%), showing distinct community characteristics (see Appendix A.2 for detailed breakdown).

Metric	**Mean**	**Range**
Q length (char)	94.4	6–2,030
Images per Q	1.3	1–6
Checklist items	3.3	1–5
Category	**# Items**	**%**
Gaming	91	13.9
Entertainment/Arts	50	7.7
Natural Objects	92	14.1
Science	81	12.4
Mathematics	26	4.0
IT/Computer	75	11.5
Coding/Development	45	6.9
Machinery	22	3.4
Daily Life	51	7.8
Business/Economics	37	5.7
Transportation	35	5.4
Shopping/Consumer	27	4.1
Health/Medical	21	3.2
Total	**653**	**100.0**

Table 1: **Overview of HAERAE-Vision.** Statistics of question length, number of images, and checklist items, highlighting the diversity and multimodal nature of HAERAE-Vision.

2.3 HAERAE-Vision-Explicit

To isolate the effect of query under-specification, we construct a parallel dataset where each question is rewritten to explicitly state the missing information while preserving the original intent. Figure 3 illustrates the transformation from under-specified to explicit queries across different domains.

We use GPT-5.1 with web search to rewrite each question following strict guidelines (Appendix B.4): (1) preserve the original intent and scope without broadening or narrowing, (2) make implicit context explicit by specifying domains, entities, and concrete references, (3) replace vague references such as “this,” “that,” or “here,” (4) incorporate visual information from the image into the question, and (5) use web search only to verify proper nouns (e.g., game titles, product names) implied by the original question. Each rewritten question then undergoes human validation. Three annotators reviewed all 653 explicitated questions against their corresponding images,

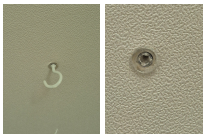
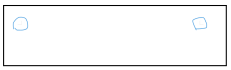
Image	Original	Explicitated
	이거는 어떻게 빼는걸까요? 저 고리를 빼고나니 저렇게 남았는데 저부분은 어떻게 빼야하나요? (How do I remove this? After removing the hook, this part remains—how do I take it out?)	천장에 설치된 흰색 고리형 행거를 제거한 후 남은 금속 부속품을 완전히 분리하려면 어떻게 해야 하나요? (How do I completely remove the metal fitting left after detaching the white ceiling hook hanger?)
	어린용 저 3마리 말고 더 있나요? (Are there more besides those 3 baby dragons?)	게임 '원신'에서 파카틴 NPC가 의뢰하는 임무 중 등장하는 이 어린 용 세 마리 외에 추가로 찾아야 하는 용이 더 있나요? (In Genshin Impact, are there additional dragons to find beyond the three baby dragons in Parkatin's quest?)
	한글 머리말 경계선 없애는 법. 동그라미 친 부분 없앨 수 있나요? (How to remove header border in Hangul. Can I remove the circled part?)	한글 문서에서 머리말 구역 상단에 표시되는 여백 경계선을 제거하려면 어떻게 해야 하나요? (How do I remove the margin border line shown at the top of the header area in Hangul word processor?)

Figure 3: **Examples of query explicitation across three domains (Daily Life, Gaming, IT/Software).** Original queries contain vague references that depend on images. Explicitated versions include background information to clarify the user request.

verifying factual accuracy, correcting hallucinated details through additional search, and adjusting specificity by removing overly specific terms or adding missing context where necessary. This process yields 653 explicitated questions paired with the original under-specified versions.

2.4 Korean Cultural Grounding

We consider an item *culturally grounded* if it requires knowledge of Korean institutions, services, policies, local brands or products, or Korean-language UI and text conventions; items solvable through globally shared knowledge are marked non-cultural. Under this criterion, 23.7% of items require distinctively Korean cultural knowledge, including local interfaces (Seoul Metro signage, Naver SmartPlace), region-specific objects (winter road sandbags), or Korean media (drama actors, traditional calligraphy). These items are rarely represented in English-centric training corpora. Figure 4 shows representative examples.

3 Evaluation Framework

3.1 Checklist-based Assessment

To mitigate the subjectivity of single-label scoring and the noise inherent in raw web text, our methodology centers on detailed checklists that decompose complex answers into specific criteria. Supported by recent findings that instance-specific rubrics align better with human judgments (Kim et al., 2024), each problem includes 1–5 evaluation points assessing different reasoning aspects. This checklist approach provides several advantages over traditional methods: (1) Fine-grained assessment of partial understanding, (2) Reduced subjectivity through explicit criteria, (3) Diagnostic capability for pinpointing model weaknesses, and (4) Scalability for automated evaluation.

3.2 LLM Judge Protocol

GPT-5-Mini is instructed to act as the primary judge, following a structured prompt that enforces consistent scoring across all problems (Appendix D). Each checklist item is scored on a three-level scale: *met* (1.0), *partially met* (0.5), or *not met* (0.0), based solely on explicit evidence found in the model's response. Each score is accompanied by supporting evidence and justification, where the evidence is a single line directly extracted from the response and the justification is a short rationale clarifying the decision. The model outputs a structured report containing evidence blocks and fractional totals (e.g., `3.5/5` when one item is partially and three are fully satisfied out of five). The overall score is computed as the average of instance-level means, where each instance has m_i checklist items with item scores $r_{ij} \in \{0, 0.5, 1\}$:

$$S_{\text{final}} = \frac{1}{N} \sum_{i=1}^{N} \left(\frac{1}{m_i} \sum_{j=1}^{m_i} r_{ij} \right),$$

ensuring comparability across problems with differing checklist lengths.

4 Experimental Setup

4.1 Model Selection

We evaluate 45 VLMs covering a broad range of families and scale. **Proprietary models.** This group includes OpenAI's GPT-5 series (`GPT-5`, `GPT-5-Mini`, `GPT-5-Nano`) (OpenAI, 2025a), Google's Gemini (2.5-Pro, 2.5-Flash, 2.5-Flash-Lite) (Google DeepMind, 2025), and proprietary

Figure 4: **Examples highlighting the cultural specificity of HAERAE-Vision:** (a) Seoul subway interface, (b) traditional painting with calligraphy, (c) Korean drama scene requiring celebrity recognition, (d) TV channel settings, (e) historical family registry. Such culturally grounded items require knowledge rarely represented in English-centric datasets.

systems such as `Perplexity-Sonar-Pro` (Perplexity AI, 2025), `xAI-Grok-4` (xAI, 2025), Mistral (`Medium-3.1`, `Small-24B`) and Pixtral (`Large`, `12B`) (Mistral AI, 2024; Agrawal et al., 2024). **Open-source models.** We evaluate `Gemma-3` (27B, 12B, 4B) (Gemma Team, Google DeepMind, 2025), `Qwen2.5-VL` (72B, 7B, 3B) (Bai et al., 2025), `Qwen3-VL` (235B-A22B, 32B, 30B-A3B, 8B, 4B, 2B; each in *Instruct* and *Thinking* variants) (Yang et al., 2025), `Skywork-R1V3-38B` (Shen et al., 2025), `InternVL3.5` (38B–1B) (Wang et al., 2025), and `AIDC-AI-Ovis2` (34B–1B) (Lu et al., 2025). **Korean models.** Finally, we include Korean-specific models, including `VARCO-VISION-2.0` (14B, 1.7B) (NCSOFT AI Center, 2025) and `HyperCLOVA-3B` (Yoo et al., 2024).

4.2 Implementation Details

We used `temperature=0.6` (1.0 for GPT-5 due to provider constraints), `top_p=0.95`, and `max_tokens=4096` across all models. Each instance was evaluated three times and averaged.

5 Results

5.1 Overall Performance

Table 2 summarizes the performance of 18 VLMs across four categories (full results are provided in Appendix E.1). Even the best-performing models—Gemini 2.5 Pro (48.5%) and GPT-5 (48.0%)—fall short of 50% accuracy, highlighting that authentic, culturally grounded multimodal queries remain far from solved. Proprietary systems consistently outperform open-weight counterparts, with the strongest open-weight models (Skywork-R1V3-38B: 27.8%, Qwen2.5-VL-72B: 20.6%) reaching roughly half the accuracy of top proprietary models. Neither search-augmented models (Perplexity Sonar-Pro: 44.3%) nor reasoning-specialized models (Skywork-R1V3) achieve notable gains, suggesting that solving would require capabilities beyond current retrieval-augmented or reasoning-optimized paradigms.

Korean-specialized models also struggled to achieve competitive results (VARCO-VISION 2.0 14B: 15.6%, HyperCLOVA X-SEED-3B: 12.7%), indicating that dedicated local models have yet to demonstrate clear advantages on this benchmark. See Appendix E for a domain-level analysis.

5.2 Effect of Query Explicitation

Figure 5 shows the effect of query explicitation on model performance. Across all six models, explicitation yields substantial improvements of 7.8 to 21.7 points. Smaller models benefit most from explicitation: GPT-5-Nano improves by 21.7 points (21.2 $\rightarrow$ 43.0), more than doubling its performance, while larger models like GPT-5 and Gemini 2.5 Pro show gains of 9.6 and 8.1 points respectively. This pattern suggests that under-specified queries disproportionately disadvantage smaller models, which may lack the capacity to infer implicit context from images alone. Even with explicitation, the best-performing model (GPT-5) achieves only 57.6%, indicating that query under-specification accounts for a substantial portion, but not all, of the difficulty in HAERAE-Vision. Our error analysis (Section 6) reveals that the remaining challenges stem primarily from cultural knowledge gaps.

5.3 Effect of Web Search

To isolate the contributions of query explicitation and retrieval augmentation, we evaluated GPT-5 and GPT-5-Mini across all four conditions: original and explicitated queries, each with and without web search. We use the official OpenAI search API (OpenAI, 2025b).

As shown in Table 3, web search yields moderate improvements for original queries (GPT-5: +7.57; GPT-5-Mini: +5.87), but these gains are smaller than those obtained through explicitation alone (+9.56 and +7.83, respectively). Notably,

Model	Entertainment	Scientific	Technical	Daily Life	Overall
Proprietary Models					
Gemini 2.5 Pro	$\mathbf{40.52_{0.61}}$	$\mathbf{51.44_{0.40}}$	$53.89_{0.79}$	$52.64_{0.93}$	$\mathbf{48.54_{0.11}}$
GPT-5	$33.07_{0.87}$	$48.14_{0.96}$	$\mathbf{55.71_{0.84}}$	$\mathbf{55.98_{0.75}}$	$48.01_{0.19}$
GPT-5 Mini	$27.38_{0.81}$	$50.62_{0.93}$	$51.88_{0.74}$	$51.31_{1.32}$	$45.21_{0.70}$
Perplexity Sonar-Pro	$32.84_{0.76}$	$47.98_{0.59}$	$47.17_{1.23}$	$49.64_{0.64}$	$44.28_{0.48}$
Gemini 2.5 Flash	$29.31_{1.09}$	$45.04_{0.98}$	$44.05_{0.53}$	$48.72_{1.38}$	$41.05_{0.79}$
Grok-4	$26.88_{0.67}$	$31.03_{0.64}$	$44.18_{0.80}$	$39.67_{0.55}$	$36.08_{0.30}$
Gemini 2.5 Flash-Lite	$18.39_{0.59}$	$38.17_{1.47}$	$32.74_{0.84}$	$35.47_{0.92}$	$30.29_{0.24}$
GPT-5 Nano	$11.64_{0.53}$	$20.10_{1.24}$	$27.15_{1.36}$	$29.68_{0.54}$	$21.22_{0.26}$
Open Source Models					
Skywork-R1V3-38B	$15.03_{0.73}$	$35.31_{0.88}$	$30.22_{0.49}$	$33.75_{0.72}$	$27.76_{0.34}$
Mistral Medium 3.1	$13.74_{0.80}$	$30.77_{0.86}$	$28.87_{0.67}$	$28.78_{1.01}$	$24.86_{0.56}$
Gemma-3 27B	$11.59_{0.58}$	$25.80_{0.61}$	$22.28_{1.04}$	$30.85_{0.61}$	$22.53_{0.16}$
Qwen2.5-VL-72B	$10.89_{0.66}$	$26.71_{1.49}$	$21.60_{0.53}$	$25.61_{0.52}$	$20.58_{0.46}$
Pixtral Large	$11.43_{0.82}$	$21.79_{0.50}$	$21.77_{0.38}$	$25.65_{0.91}$	$20.10_{0.24}$
InternVL3.5-38B	$8.81_{0.46}$	$23.25_{0.61}$	$17.92_{0.73}$	$23.36_{0.78}$	$18.01_{0.22}$
Ovis2-34B	$9.52_{0.47}$	$21.88_{0.55}$	$21.00_{0.51}$	$24.82_{0.58}$	$18.50_{0.02}$
Mistral Small 24B	$6.46_{0.29}$	$10.18_{0.45}$	$13.30_{0.66}$	$16.20_{0.66}$	$11.20_{0.01}$
Korean-specialized Models					
VARCO-VISION 2.0 (14B)	$7.87_{0.80}$	$16.56_{0.65}$	$16.88_{0.57}$	$22.13_{0.88}$	$15.55_{0.29}$
HyperCLOVA X-SEED-3B	$6.25_{0.25}$	$14.87_{0.51}$	$11.99_{0.50}$	$17.93_{0.73}$	$12.66_{0.10}$

Table 2: **Performance of 18 models averaged by category.** For model families with multiple sizes, only the largest variant is shown. Full results across all model sizes and detailed 13-category breakdowns are in Appendix 15. All scores are reported as mean_{SE}, where SE is the standard error over 3 independent runs (n=3). The highest-scoring model is highlighted in **bold**.

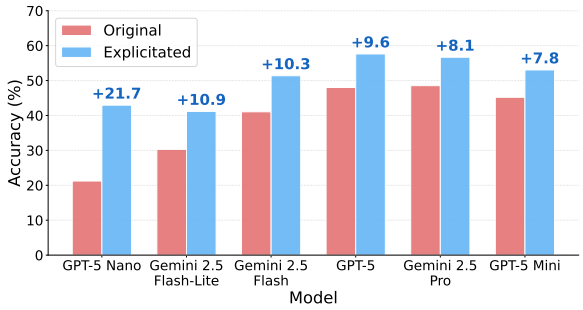


Figure 5: **Effect of query explicitation on model performance.** Models are sorted by improvement magnitude. Smaller models benefit most from explicitation, with GPT-5 Nano showing +21.7 points improvement. All results averaged over 3 runs.

Model	Orig	Orig+S	Expl	Expl+S
GPT-5	48.01	55.58	57.57	59.72
GPT-5-Mini	45.21	51.08	53.04	56.69
Δ *from Original (no search)*				
GPT-5	–	+7.57	+9.56	+11.71
GPT-5-Mini	–	+5.87	+7.83	+11.48

Table 3: **Effect of web search and query explicitation.** Scores reported as mean over 3 runs. Original+Search still underperforms Explicit alone, indicating retrieval cannot compensate for under-specification.

original queries augmented with search still underperform explicit queries without search (GPT-5: 55.58 vs. 57.57; GPT-5-Mini: 51.08 vs. 53.04). This indicates that retrieval cannot compensate for under-specified queries; models must first infer user intent for search to be effective. We observe a recurring failure mode in which models rely on textual cues during search while failing to ground visual features, suggesting that current web search integration operates at a largely surface level and is not deeply leveraged by GPT-5. The highest performance is achieved when explicitation and search are combined (GPT-5: 59.72; GPT-5-Mini: 56.69), demonstrating additive benefits. However, the marginal improvement from adding search to explicit queries (+2.15 and +3.65) is smaller than when added to original queries, implying that explicitation already supplies much of the contextual information that search would otherwise retrieve.

5.4 Cross-Lingual Validation

To test whether the explicitation effect generalizes beyond Korean, we conduct a pilot study in English. We collect approximately 3,000 image-containing Q&A pairs from 12 Stack Exchange communities spanning 9 of our 13 categories and apply the same six-stage pipeline (Section 2.1). After filtering, 627 samples survive; we randomly select 100 samples stratified by category for evaluation (see Appendix A.3 for full construction details and per-domain results).

As shown in Table 4, all four models show consistent explicitation gains (+3.2 to +6.6 points), confirming that the effect of query underspecification is not limited to Korean. However, English deltas are consistently smaller than their Korean counterparts (+7.8 to +21.7). Notably, GPT-5-Nano scores 44.4% on English original queries, more

Model	Original	Explicit	Δ
GPT-5	60.8	65.6	+4.8
GPT-5-Mini	53.3	59.9	+6.6
Gemini 2.5 Pro	51.3	57.3	+6.0
GPT-5-Nano	44.4	47.6	+3.2

Table 4: **English pilot: effect of query explicitation.** All four models show consistent gains, confirming that underspecification effects are cross-lingual. Deltas are smaller than in Korean (+3.2–6.6 vs. +7.8–21.7).

than double its Korean score (21.2%), suggesting that smaller models can better compensate for underspecification in high-resource languages.

Root-cause analysis on remaining errors after explicitation reveals why the gap is smaller in English: cultural knowledge accounts for only 2.7% of English errors versus 19.0% in Korean, with general reasoning comprising 96.7% of English failures. This confirms that the larger Korean explicitation gap is driven by the interaction between surface-level underspecification and culturally grounded assumptions that are underrepresented in English-centric training corpora. Once explicitation resolves surface ambiguity, cultural knowledge remains as a persistent source of difficulty in Korean but not in English.

6 Additional Analysis on Explicitation

To understand why explicitation improves performance, we analyzed error patterns across original and explicitated conditions. We collected 3,164 (original) and 2,834 (explicitated) error cases where models scored below 1.0, spanning six models (GPT-5, GPT-5-Mini, GPT-5-Nano, Gemini 2.5 Pro, Gemini 2.5 Flash, Gemini 2.5 Flash-Lite). Each error was annotated by an LLM judge (Claude 3.5 Sonnet) along two dimensions: (1) *failure category*—how the error manifests (Table 5); and (2) *root cause*—why the error occurs. Failure categories are multi-label (1–3 per error), while root causes are single-label; the two dimensions are orthogonal, so their percentages are not directly comparable. The full annotation prompt and category definitions are provided in Appendix F.

6.1 What Explicitation Fixes

Table 6 shows the key shifts. The most striking change is the reduction in *lack of explicitness* failures, which drop from 84.3% to 69.7% (-14.6pp), directly confirming that explicitation addresses surface-level ambiguity. Smaller models show the largest reductions in error cases after explicitation (GPT-5-Nano: -83 cases, +12.7pp perfect rate) compared to larger models (GPT-5-Mini: -40 cases, +6.1pp), confirming that under-specification disproportionately impacts smaller models.

Failure Category	Description
Lack of explicitness	Missing checklist-required facts
Procedural reasoning	Failed multi-step execution
Object recognition	Misidentified visual entities
Cultural mismatch	Misunderstood Korean conventions
Visual-text grounding	Wrong image region referenced
Spatial reasoning	Incorrect spatial relations
Root Cause	
General reasoning	Logic/inference failure
Cultural knowledge	Missing Korean-specific knowledge
Language	Korean language misunderstanding

Table 5: **Error annotation taxonomy (abbreviated).**

Failure Category	Orig	Expl	Δ
Lack of explicitness	84.3%	69.7%	-14.6
Procedural reasoning	66.6%	64.3%	-2.3
Object recognition	20.6%	18.5%	-2.1
Cultural concept mismatch	13.1%	22.5%	+9.4
Visual-text grounding	5.2%	16.6%	+11.4

Table 6: **Failure category shifts from original to explicitated queries.**

Category-level analysis (Figure 6) reveals that explicitation yields the largest gains in Mathematics, Science, Coding, and Shopping—categories where failures primarily stemmed from underspecified problem descriptions. In contrast, Natural Objects and Entertainment remain challenging even after clarification (all-models-pass rate: 0% in both conditions), with failures shifting toward visual-text grounding and cultural knowledge gaps.

Notably, visual-text grounding (VTG) errors increase from 5.2% to 16.6% after explicitation. However, tracking individual error cases reveals that this reflects an *unmasking* effect rather than a trade-off: 87% of VTG errors in the explicitated condition were already errors under original queries but classified under other categories (primarily lack of explicitness). Explicitation forces models to engage with specific visual regions, exposing latent grounding failures previously obscured by surface-level ambiguity (see Appendix E.5 for detailed analysis).

6.2 Why Retrieval Alone Is Insufficient

Earlier, our results have shown that original queries with search (55.6) underperform explicit queries without search (57.6). This reveals a fundamental limitation: retrieval cannot compensate for query

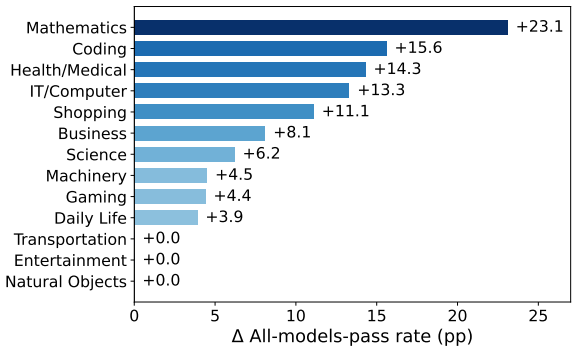


Figure 6: **Category-level explicitation effects.** Categories like Mathematics and Coding show large gains, while Entertainment and Natural Objects remain difficult even after clarification, with failures shifting toward cultural knowledge and visual grounding.

under-specification. Under-specified queries like "이거 어떻게 해요?" (How do I do this?) contain no searchable keywords. Since the critical context is embedded solely within the visual modality, current text-based search engines fail to bridge the modality gap without explicit textual grounding. Even when models attempt searches, they lack the specific terms (product names, game titles, error codes) needed to retrieve useful results. In contrast, explicitated queries contain concrete references (e.g., "천장에 설치된 흰색 고리형 행거" (white ring-shaped hanger installed on the ceiling)) that enable targeted retrieval. The best performance is achieved when both are combined (59.7), but the key finding is that search on under-specified queries cannot match explicitation alone; models must first understand what to search for.

6.3 Cultural Knowledge Gaps

After explicitation, what errors remain? Analyzing root causes reveals a shift toward cultural knowledge gaps (Table 7). The increase in cultural knowledge attribution (+6.4pp) suggests that once query ambiguity is resolved, the dominant remaining challenge is Korea-specific knowledge. For example, when shown orange bags along a rural road, models identified them as "road safety markers" or "wasp traps," missing that these are winter snow preparation sandbags, something all native Korean drivers would have known. Similarly, all SOTA models misidentified a Korean folder phone (SKY IM-100) as global brands like Sony or Nokia. Finally, the negligible language error rate (<1.5%) confirms that Korean proficiency is no longer a hurdle for global models, but cultural content is.

Root Cause	Orig	Expl	Δ
General reasoning	86.6%	79.8%	-6.9
Cultural knowledge	12.7%	19.0%	+6.4
Language	0.7%	1.2%	+0.5

Table 7: **Root cause distribution.** After explicitation, cultural knowledge becomes more prominent as surface-level ambiguity is resolved.

	GPT-5-mini	GPT-5	Gem-2.5-Pro	Gem-2.5-Flash
GPT-5-mini	–	0.87	0.90	0.90
GPT-5	0.87	–	0.90	0.86
Gem-2.5-Pro	0.90	0.90	–	0.89
Gem-2.5-Flash	0.90	0.86	0.89	–
Krippendorff's $\alpha = 0.867$				

Table 8: **Pairwise Pearson correlations among four LLM judges.** Spearman correlations range 0.87–0.90. Krippendorff's $\alpha = 0.867$ indicates substantial agreement.

7 Reliability of LLM-as-a-Judge

It is widely known that LLM-Judges may be prone to biases (Son et al., 2024). Accordingly, to ensure the credibility of our evaluation, we assess the inter-judge agreement among four LLM judges (GPT-5, GPT-5-mini, Gemini-2.5-Pro, Gemini-2.5-Flash). A stratified random sample of 250 model responses (50 per 0.2-score interval) was re-evaluated under identical protocols. Table 8 shows consistently high correlations, with Pearson ranging from 0.863 to 0.903 and Spearman from 0.866 to 0.901. Krippendorff's $\alpha = 0.867$ exceeds the conventional 0.80 threshold, indicating substantial agreement across models with different architectures.

Furthermore, to assess alignment with human judgments, the same 250-sample set was evaluated by four independent human annotators, who rated the appropriateness of GPT-5-Mini judgments on a 5-point scale. Agreement was high (Pearson $r = 0.820$, Spearman $\rho = 0.810$, $p < 0.001$), demonstrating that our judge provides a stable and human-aligned evaluation signal. Detailed analyses of low-agreement cases suggest that most discrepancies stem from superficial keyword matching or excessive leniency (examples in Appendix C.2).

8 Related Work

Evaluating VLMs. As VLMs become more general-purposed, evaluation has shifted toward diagnostic suites that aim to separate recognition, OCR, and knowledge from higher-level reasoning and instruction following (Liu et al., 2024; Li et al., 2024; Yu et al., 2024). To better probe reasoning, several benchmarks target domain knowl-

edge grounded with visual inputs (Yue et al., 2024, 2025; Lu et al., 2023). This was rapidly followed by the Korean community, first by text benchmarks that measure Korean knowledge (Son et al., 2023, 2025; Hong et al., 2025), then by multimodal benchmarks: KRETA, KViscuit, and KOFFVQA (Hwang et al., 2025; Park et al., 2024; Kim and Jung, 2025). In addition, localized evaluation tools such as KMMB, KSEED, and KDTCBench have been released alongside Korean VLM development efforts (Ju et al., 2024). However, these benchmarks have already been saturated by older-generation models such as GPT-4o (e.g., KRETA (Hwang et al., 2025): 84.6; K-VISCUIT (Park et al., 2024): 89.5; K-MMB: 81.01; K-SEED: 76.98; K-DTCBench: 85.80 (Ju et al., 2024)), motivating the creation of a more challenging benchmark.

Query Underspecification. Underspecified or ambiguous queries are pervasive in conversational settings (Rahmani et al., 2023), forcing systems to choose between answering, hedging, or asking for missing constraints. Prior efforts to evaluate LLMs in ambiguity handling include AmbigQA (Min et al., 2020), and clarification-focused resources such as ClariQ (Aliannejadi et al., 2021) and the ConvAI3 shared task (Aliannejadi et al., 2020), which measure how effectively a system reduces uncertainty through clarification. More recently, QuestBench tests minimal question asking as information acquisition for underspecified reasoning (Li et al., 2025). In the multimodal setting, ClearVQA evaluates whether models can ask image grounded clarification questions to resolve ambiguous visual queries (Jian et al., 2025). Overall, however, multimodal resources for query underspecification remain scarce. To bridge this gap, we introduce HAERAE-Vision, which further targets a niche and underexplored setting by focusing on underspecification in Korean language interactions with culturally grounded content and assumptions.

9 Conclusion

We introduce HAERAE-Vision, a benchmark of 653 authentic Korean questions from real-life users, each paired with explicit rewrites. Our experiments show that query underspecification accounts for an 8–22 point drop in VLM performance. Retrieval-augmented prompting does not close this gap: search-augmented underspecified queries still underperform explicitated queries without search. We further find that many remaining failures reflect missing cultural knowledge rather than surface-level ambiguity. An English pilot study confirms that the explicitation effect generalizes cross-lingually, with smaller deltas attributable to fewer cultural knowledge barriers. Together, these findings highlight challenges that sanitized, clean-query benchmarks fail to capture.

Limitations

Guided by a quality over quantity principle, our filtering procedure yields a 0.76% survival rate. This aggressive filtering may exclude some informative edge cases; however, it should be noted that our goal is not to provide a comprehensive evaluation of Korean knowledge. Rather, we aim to study how LLM behavior changes under different levels of information density in user prompts. Furthermore, our web search augmentation analysis is also limited in scope, as it evaluates only OpenAI's web search, and results may differ with more advanced retrieval systems. However, based on our observations, the primary bottleneck appears to be less about the search API itself and more about the model's ability to extract and formulate meaningful questions grounded in the image and accompanying text. Our error annotation relies on an LLM judge, which may introduce systematic biases despite the high inter-judge agreement we observe. Finally, while our English pilot (Section 5.4) confirms that the explicitation effect generalizes beyond Korean, it is limited to 100 samples from a single platform ecosystem; a broader multilingual investigation across diverse languages and cultural contexts remains for future work.

Ethics and Data Governance

This study received ethical approval from the Institutional Review Board of Hankuk University of Foreign Studies (HUFS-2510-015). All data were collected from publicly available Korean community platforms. We implemented a rigorous filtering process to exclude sensitive content, and all PII has been systematically removed. AI assistants (Claude and Gemini) were used for grammar editing and code debugging.

Acknowledgments

This work was supported by the Hankuk University of Foreign Studies Research Fund (2025) and the TIPS Program (No. RS-2024-00512659) funded by the Ministry of SMEs and Startups (MSS), Korea.

References

Pravesh Agrawal, Szymon Antoniak, Emma Bou Hanna, Baptiste Bout, Devendra Chaplot, and 1 others. 2024. Pixtral 12b. *arXiv*.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2020. Convai3: Generating clarifying questions for open-domain dialogue systems (clariq). *arXiv preprint arXiv:2009.11352*.

Mohammad Aliannejadi, Julia Kiseleva, Aleksandr Chuklin, Jeff Dalton, and Mikhail Burtsev. 2021. Building and evaluating open-domain dialogue corpora with clarifying questions. In *EMNLP*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Gemma Team, Google DeepMind. 2025. Gemma 3 technical report. *arXiv*.

Google DeepMind. 2025. Gemini 2.5 pro: Model card. https://storage.googleapis.com/model-cards/documents/gemini-2.5-pro.pdf.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Seokhee Hong, Sunkyoung Kim, Guijin Son, Soyeon Kim, Yeonjung Hong, and Jinsik Lee. 2025. From kmmlu-redux to kmmlu-pro: A professional korean benchmark suite for llm evaluation. *arXiv preprint arXiv:2507.08924*.

Taebaek Hwang, Minseo Kim, Gisang Lee, Seonuk Kim, and Hyunjun Eun. 2025. Kreta: A benchmark for korean reading and reasoning in text-rich vqa attuned to diverse visual contexts. *arXiv preprint arXiv:2508.19944*.

Pu Jian, Donglei Yu, Wen Yang, Shuo Ren, and Jiajun Zhang. 2025. Teaching vision-language models to ask: Resolving ambiguity in visual questions. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3619–3638, Vienna, Austria. Association for Computational Linguistics.

Jeongho Ju, Daeyoung Kim, SunYoung Park, and Youngjune Kim. 2024. Varco-vision: Expanding frontiers in korean vision-language models. *arXiv preprint arXiv:2411.19103*.

Seungone Kim, Juyoung Suk, Ji Yong Cho, Shayne Longpre, Chaeeun Kim, Dongkeun Yoon, Guijin Son, Yejin Cho, Sheikh Shafayat, Jinheon Baek, and 1 others. 2024. The biggen bench: A principled benchmark for fine-grained evaluation of language models with language models. *arXiv preprint arXiv:2406.05761*.

Yoonshik Kim and Jaeyoon Jung. 2025. Koffvqa: An objectively evaluated free-form vqa benchmark for large vision-language models in the korean language. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 575–585.

Belinda Z Li, Been Kim, and Zi Wang. 2025. Questbench: Can llms ask the right question to acquire information in reasoning tasks? *arXiv preprint arXiv:2503.22674*.

Bohao Li, Yuying Ge, Yixiao Ge, Guangzhi Wang, Rui Wang, Ruimao Zhang, and Ying Shan. 2024. Seedbench: Benchmarking multimodal large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13299–13308.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Shiyin Lu, Yang Li, Yu Xia, Yuwei Hu, Shanshan Zhao, Yanqing Ma, Zhichao Wei, Yinglun Li, Lunhao Duan, Jianshan Zhao, Yuxuan Han, Haijun Li, Wanying Chen, Junke Tang, Chengkun Hou, Zhixing Du, Tianli Zhou, Wenjie Zhang, Huping Ding, and 23 others. 2025. Ovis2.5 technical report. *Preprint*, arXiv:2508.11737.

Sewon Min, Julian Michael, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2020. Ambigqa: Answering ambiguous open-domain questions. *arXiv preprint arXiv:2004.10645*.

Mistral AI. 2024. Pixtral-large-instruct-2411: Model card. https://huggingface.co/mistralai/Pixtral-Large-Instruct-2411.

NCSOFT AI Center. 2025. Varco-vision-2.0 technical report. *arXiv*.

OpenAI. 2025a. Gpt-5 system card. https://openai.com/index/gpt-5-system-card/. Updated PDF: https://cdn.openai.com/gpt-5-system-card-aug7.pdf.

OpenAI. 2025b. Web search — openai api reference.

ChaeHun Park, Yujin Baek, Jaeseok Kim, Yu-Jung Heo, Du-Seong Chang, and Jaegul Choo. 2024. Evaluating visual and cultural interpretation: The k-viscuit benchmark with human-vlm collaboration. *arXiv preprint arXiv:2406.16469*.

Perplexity AI. 2025. Sonar pro: Model overview. https://docs.perplexity.ai/getting-started/models/models/sonar-pro.

Hossein A Rahmani, Xi Wang, Yue Feng, Qiang Zhang, Emine Yilmaz, and Aldo Lipani. 2023. A survey on asking clarification questions datasets in conversational systems. *arXiv preprint arXiv:2305.15933*.

W. Shen and 1 others. 2025. Skywork-r1v3 technical report. *arXiv*.

Guijin Son, Hyunwoo Ko, Hoyoung Lee, Yewon Kim, and Seunghyeok Hong. 2024. Llm-as-a-judge & reward model: What they can and cannot do. *arXiv preprint arXiv:2409.11239*.

Guijin Son, Hanwool Lee, Sungdong Kim, Seungone Kim, Niklas Muennighoff, Taekyoon Choi, Cheonbok Park, Kang Min Yoo, and Stella Biderman. 2025. Kmmlu: Measuring massive multitask language understanding in korean. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4076–4104.

Guijin Son, Hanwool Lee, Suwan Kim, Huiseo Kim, Jaecheol Lee, Je Won Yeom, Jihyu Jung, Jung Woo Kim, and Songseong Kim. 2023. Hae-rae bench: Evaluation of korean knowledge in language models. *arXiv preprint arXiv:2309.02706*.

Weiyun Wang and 1 others. 2025. Internvl 3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv*.

xAI. 2025. Grok 4: Model card. https://data.x.ai/2025-08-20-grok-4-model-card.pdf.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Kyung-Min Yoo and 1 others. 2024. Hyperclova x technical report. *arXiv*.

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2024. Mm-vet: Evaluating large multimodal models for integrated capabilities. In *International conference on machine learning*. PMLR.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, and 1 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, and 1 others. 2025. 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)*, pages 15134–15186.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Appendices

A Dataset Construction Details

A.1 Detailed Platform Descriptions

We collected data from nine Korean online platforms representing diverse user communities and domain expertise. Table 10 provides detailed information about each platform.

Platform	Category	Description
Naver KnowledgeIn	General Q&A	Korea's largest general Q&A platform covering everyday queries, academic subjects, and technical issues
BRIC	Science Community	Specialized community for biological research and biotechnology with scientific discussions and professional knowledge sharing
Ruliweb	Gaming Community	Major gaming community covering video games, hardware reviews, game mechanics, and technical gaming issues
MonsterZym	Fitness Community	Fitness and bodybuilding community discussing workout routines, nutrition, supplements, and exercise techniques
Quasarzone	Hardware Community	Hardware enthusiast community focused on computer components, electronics, PC building, and technology reviews
i-Boss	Business Platform	Business and entrepreneurship platform for startup strategies, operations, marketing, and professional development
Inflearn	Coding Education	Online learning platform with community features for programming questions and coding experiences
Codeit	Coding Education	Coding education platform with forums for programming discussions and technical support
Okky	Developer Community	Developer community platform for programming discussions, career advice, and technical problem-solving

Table 9: Korean online platforms used for data collection

These platforms were selected to ensure comprehensive coverage of different user demographics, expertise levels, and domain-specific knowledge, reflecting the diversity of real-world multimodal questions Korean users encounter online.

A.2 Platform-wise Filtering Statistics

Table 10 provides a detailed breakdown of data collection and filtering across all platforms.

Platform	Raw Data	Appropri.	Difficulty	Image Dep.	Human Val.	Final	Survival
KnowledgeIn	31,484	10,495	1,404	648	441	441	1.4%
BRIC	291	291	163	60	42	42	14.4%
Ruliweb	305	240	54	42	32	32	10.5%
Coding	27,896	8,369	837	198	135	135	0.5%
MonsterZym	3,090	3,090	2,234	8	6	6	0.2%
Quasarzone	2,986	896	90	22	15	15	0.5%
i-Boss	20,000	20,000	578	62	42	42	0.2%
Total	86,052	43,381	5,360	1,040	713	653	0.76%

Table 10: Detailed data collection and filtering statistics by platform (Stages 1–6). Coding platforms include Inflearn, Codeit, and Okky combined.

A.3 English Pilot Study Details

To validate the cross-lingual generalizability of our findings, we construct an English pilot dataset by applying the same six-stage pipeline described in Section 2.1.

Data Source. We collect 2,954 image-containing Q&A pairs from 12 Stack Exchange communities: Stack Overflow, Super User, Arqade, DIY Home Improvement, Biology, Gardening & Landscaping, Motor Vehicle Maintenance, Cooking, Bicycles, Chemistry, Board & Card Games, and Mathematics. All data are publicly available under CC BY-SA 4.0.

Filtering and Explicitation. All pipeline stages are applied without modification. Table 11 summarizes the filtering process. Image dependency verification is the most aggressive filter, removing over 60% of candidates where images served as supplementary illustration rather than essential context. Each surviving question is explicitated following the same protocol as the Korean dataset (Appendix B.4), with subsequent human verification. We randomly select 100 samples stratified by domain for the pilot evaluation.

Stage	Remaining	Removed
Raw collection	2,954	–
Image validation	2,876	−78
Under-specification filter	2,457	−419
Difficulty calibration	2,326	−131
Image dependency verification	887	−1,439
Checklist + explicitation	887	–
Quality filtering	627	−260
Stratified sampling	100	–

Table 11: **English pilot filtering pipeline.**

Dataset Statistics. The final 100 samples contain 168 images (avg 1.7 per question) and an average of 3.4 checklist items. Table 12 shows the domain distribution.

Domain	# Items	%
Natural Objects / Science	23	23.0
Transportation	18	18.0
Gaming / Entertainment	13	13.0
Daily Life / Machinery	12	12.0
Coding	11	11.0
IT / Computer	9	9.0
Daily Life	7	7.0
Science	4	4.0
Mathematics	3	3.0
Total	**100**	**100.0**

Table 12: **Domain distribution of the English pilot dataset.**

Per-Domain Results. Table 13 presents per-domain explicitation effects averaged across all four models. Consistent with the Korean results, Coding shows the largest gain (+14.4), while visually grounded domains such as Natural Objects show smaller improvements.

Domain	n	Orig	Expl	Δ
Coding	11	57.5	71.8	+14.4
Gaming / Ent.	13	40.0	49.0	+9.0
Daily Life	7	47.3	55.6	+8.3
Science	4	49.8	57.8	+8.0
Transportation	18	51.1	56.5	+5.4
Nat. Objects / Sci.	23	55.4	58.3	+2.9
IT / Computer	9	55.4	58.1	+2.7
Mathematics	3	49.4	50.6	+1.1
Daily Life / Mach.	12	60.1	56.8	−3.3

Table 13: **English pilot: per-domain explicitation effects.** Scores averaged across all four evaluated models.

B Pipeline Prompts

B.1 Stage 2 (Safety, Objectivity, Temporal)

We used three LLM-based filters in Stage 2: content safety, objectivity, and temporal dependency. Below we excerpt only the core exclusion criteria from the prompts (full wording omitted).

B.1.1 Content Safety

Mark as inappropriate if the question–image pair includes:

- Political content (politicians, parties, elections, political opinions)
- Religious advocacy/criticism or conflicts
- Hate/discrimination
- Suicide or self-harm; sensitive mental-health topics
- Sexual/adult content, nudity, explicit innuendo

B.1.2 Objectivity

Mark as inappropriate if the pair is subjective or ambiguous, e.g.:

- Preference/aesthetic judgments ("pretty/ugly", "which outfit is nicer?")
- Suitability/personal advice without criteria
- Moral/intentionality speculation ("who is wrong?", "good person?")
- Multiple valid interpretations or unverifiable answers

B.1.3 Temporal Dependency

Mark as inappropriate if the pair requires time-specific information, e.g.:

- "today/now" weather, traffic, store hours, last train
- Current events or status queries ("is it open now?", "stock price today?")
- Questions that become invalid/meaningless as time passes

B.2 Stage 4 Prompt Excerpt (Image Dependency Rubric)

Input: (Q), model answer with image, model answer without image, optional gold answer snippet.
Task: Compare the two answers and decide image dependency.
Decision labels

- IMAGE_REQUIRED: with-image answer is substantially more accurate/specific; text-only answer is vague, incorrect, or explicitly requests the image.
- NO_IMAGE_NEEDED: both answers are comparable in correctness and specificity without relying on visual cues.
- UNCERTAIN: evidence is inconclusive (e.g., partial improvements or conflicting signals).

Scoring (1–5 quality gap)

- 1: negligible difference; 3: clear but moderate gain; 5: decisive gain (critical visual details).

Output (natural language)

- `Judgment:` IMAGE_REQUIRED / NO_IMAGE_NEEDED / UNCERTAIN
- `Reason:` brief comparison citing concrete differences
- `QualityGap:` integer in {1,2,3,4,5}

B.3 Stage 5 (Checklist Generation)

This appendix provides the instruction prompt used for checklist generation along with illustrative examples of the resulting decompositions. We used GPT-4-mini to derive structured criteria directly from reference answers that users found satisfactory. These checklists therefore represent strict, human-aligned evaluation standards: a model must satisfy all listed criteria to be considered correct.

Game (Stardew Valley)
"What is the circled item in the screenshot?"
- Identify circled item as a sap tap (수액채취기)
- Mention install only on fully grown trees
- Explain how to obtain/craft it
- Note sap can be collected after time

Economics/Management
"Cost allocation: is S2 missing 100,000?"
- Provide correct S1/S2 values
- Reset self-allocation entries to zero
- Derive allocation ratios (0.5F, 0.4M)

Daily Life
"Is this ceiling tile asbestos?"
- Identify material as gypsum, not asbestos
- Explain gypsum board contains no asbestos
- Explicitly name "석고텍스"
- Assure user it is safe

Science
"Why does neutron mass ratio decrease?"
- Explain neutron beta decay
- Clarify neutrons inside He nucleus
- Relate x-axis to cosmic cooling
- Interpret H:He ratio $\approx$ 3:1

Figure 7: Examples of checklist decomposition across domains, generated in Stage 5. For brevity, the checklists shown here are abbreviated; full checklists typically contain 1–5 criteria per item.

B.4 Query Explicitation Prompt

The following prompt was used with GPT-5.1 (web search enabled) to generate explicitated versions of under-specified queries.

You rewrite incomplete, ambiguous, or context-dependent questions into clear, fully self-contained questions. Your goal is to produce a rewritten question that can be understood and answered on its own, without requiring prior conversation or hidden context. Preserve the original intent, scope, and tone of the question. Do NOT answer the question.
Rules:
1. Intent and scope preservation
- Preserve what the original question is asking and its level of specificity.
- Do not broaden or narrow the scope.
- Do not generalize away concrete entities or situations implied by the original question or answer.

2. Essential context inclusion
- Explicitly include essential context if it is implied or required to understand the question, such as: the relevant domain or subject; the specific scenario, task, or situation involved; named entities (e.g., people, organizations, characters, locations); concrete objects, items, or targets referenced.
- Avoid vague references such as "this," "that," "here," "the scene," or "the above."

3. Search usage
- You may use search ONLY to identify widely accepted proper nouns (e.g., titles, names, commonly used labels) that are strongly implied by the original question or associated answer.
- Do NOT use search to introduce new mechanics, steps, conditions, quantities, or interpretations.
- Do NOT resolve ambiguity by inventing details.

4. Handling missing or ambiguous information
- If critical context cannot be inferred with high confidence, include a brief clarifying placeholder inside the question, such as: [SPECIFY: missing detail].
- Do not attempt to guess or "fix" the question beyond what the inputs support.

5. Image usage (if an image is provided)
- You may incorporate information visible in the image to clarify the question.
- The rewritten question must remain answerable without viewing the image.
- Do not exhaustively describe the image or convert all visual details into text.
- Include only visual information that is essential to understanding the question.

6. Language and style
- Maintain a tone consistent with the original question.
- Do not unnecessarily formalize or casualize the language.
- Remove slang, conversational fillers, and vague references that reduce clarity.

Output requirements: Output ONLY the rewritten question text. No explanations, no bullet points, no headers. Do not include meta-instructions or commentary.

C Human Annotation

C.1 Annotation Guidelines

Seven Korean-speaking annotators conducted human validation in three phases using custom web-based tools.

C.1.1 Phase 1: Conservative Filtering

Using the annotation interface shown in Figure 8, annotators independently reviewed each item along five dimensions, removing any item flagged by at least one annotator:

- **Image-Question Relevance**: Assess whether images provide essential visual information required to answer the question.
- **Question-Answer Quality**: Evaluate question clarity, answerability, and reference answer accuracy.
- **Checklist Validation**: Review each LLM-generated checklist item for necessity, clarity, and completeness.
- **Category Appropriateness**: Verify correct classification into one of 13 domain categories.
- **Overall Assessment**: Flag items with fundamental issues such as inappropriate content or unsolvable questions.

C.1.2 Phase 2: Refinement

Three annotators refined surviving items through a separate annotation interface:

- **Question Rewriting**: Rewrite unclear or ambiguous questions while preserving original intent and scope.
- **Checklist Revision**: Evaluate each LLM-generated checklist item for appropriateness, revising unclear criteria or removing items not grounded in the original question–image pair.
- **Category Re-assignment**: Re-assign categories where the original classification was incorrect, with option to propose new categories.

C.1.3 Phase 3: Final Audit

One senior annotator consolidated categories across the dataset and verified cross-item consistency.

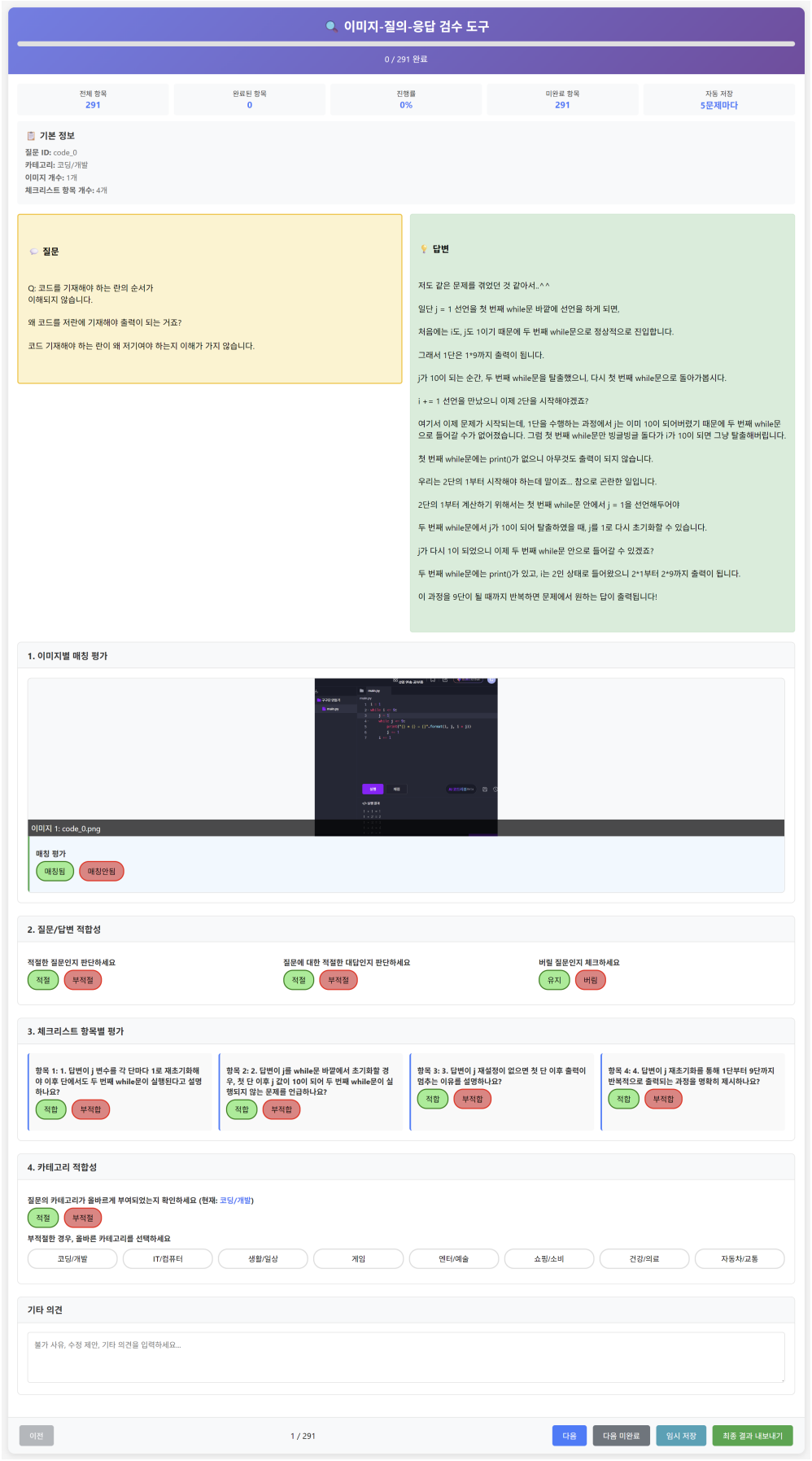


Figure 8: Screenshot of our Phase 1 annotation tool. The interface (shown in Korean) allowed annotators to assess image relevance, question/answer appropriateness, checklist accuracy, and category assignment.

C.2 LLM Judge Failure Cases

Table 14 presents representative examples of human annotator feedback for inappropriate judge evaluations, revealing systematic failure patterns.

Rating	Human Reasoning (translated)
Very Inappropriate	"Judge awarded points based on superficial word matching rather than actual checklist compliance"
Inappropriate	"Judge gave 1 point despite response not addressing checklist criteria, incorrectly interpreting explicit mention as meeting requirements"
Inappropriate	"Checklists 1,2,4 satisfied. Item 3 not clearly inappropriate but ambiguous and open to interpretation"
Inappropriate	"Even if intent aligns with checklist, response lacks clarity and remains ambiguous"
Inappropriate	"Judge overlooked insufficient explanations that clearly failed checklist requirements"

Table 14: Representative human feedback explaining inappropriate judge ratings.

Analysis reveals judge failures primarily stem from: (1) superficial keyword matching without semantic understanding, (2) excessive leniency toward incomplete responses, and (3) difficulty distinguishing between implicit intent and explicit satisfaction of requirements.

D LLM-as-Judge Prompt

This appendix provides the full prompt used for the checklist-based evaluation by the GPT-5-Mini judge. The prompt enforces explicitness, evidence grounding, and consistent scoring across items. For reproducibility, we include the full decision rules, evidence policy, and output format constraints.

[GOAL] Given a Question, Response, and a natural-language Checklist, decide for each checklist item whether the Response explicitly satisfies it: met = 1, partially met = 0.5, not met = 0. Final score = (# met) / (total checklist items).

[INPUT]

[Question] QUESTION

[Response] RESPONSE

[Checklist] CHECKLIST Treat each string as one criterion. Remove numbering such as "1." or "2)".

[DECISION RULES]

- **Use only the Response text.** No outside knowledge or assumptions. If uncertain $\rightarrow$ 0.
- **Explicitness:** direct fulfillment = 1, implicit or suggestive = 0.5, otherwise = 0.
- **Completeness ("all / every / complete"):** explicit = 1, implied = 0.5, absent = 0.
- **Method requirements:** actionable steps = 1, vague = 0.5, absent = 0.
- **"Various / multiple types":** $\geq$ 2 specific types = 1, vague or 1 type = 0.5, none = 0.
- **Synonyms:** unambiguous = 1, ambiguous = 0.5, different meaning = 0.

[EVIDENCE POLICY]

- For **1** or **0.5**: include a 10–60 character direct quote.
- For **0**: provide a brief explanation.
- Each item must include: *evidence* $\rightarrow$ *explanation* $\rightarrow$ *met*.

[OUTPUT FORMAT — STRICT]
<evidence>
Item 1:
evidence: "... direct quote from Response..."

explanation: Brief justification referencing criteria
met: 0 | 0.5 | 1

Item 2:
evidence: "..."
explanation: ...
met: 0 | 0.5 | 1

... (repeat)
</evidence>

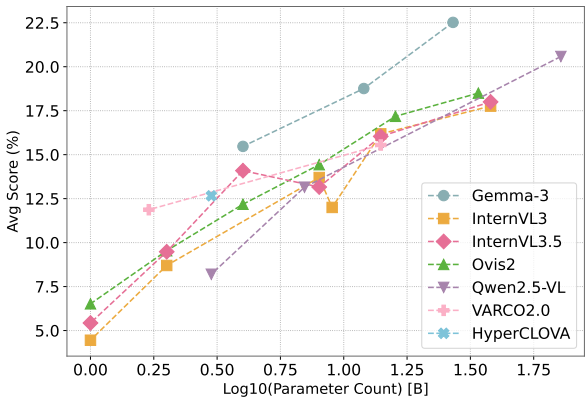
<score>
K/N
</score>

[NOTES]

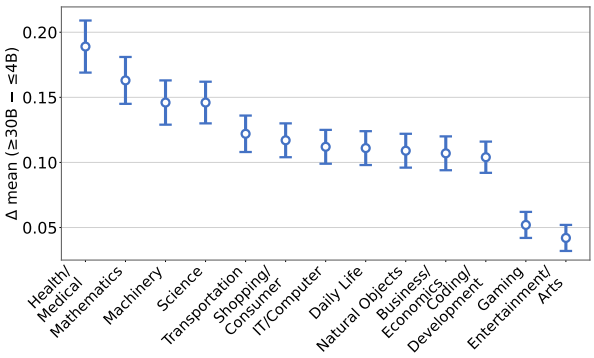
- Output only the two blocks above.
- No code fences or additional prose.

Model	IT	Health	Game	Econ	Sci	Mach	Daily	Shop	Math	Ent	Trans	Nature	Code	Avg
Proprietary Models														
Gemini 2.5 Pro	$50.73_{1.63}$	$62.17_{2.33}$	$36.67_{1.75}$	$51.09_{1.72}$	$39.93_{1.43}$	$56.22_{4.32}$	$51.32_{2.57}$	$46.00_{5.15}$	$60.94_{1.36}$	$44.37_{1.18}$	$57.69_{2.81}$	$53.45_{0.57}$	$50.91_{0.92}$	$48.54_{0.18}$
Gemini 2.5 Flash	$42.98_{0.34}$	$56.10_{3.95}$	$26.70_{1.04}$	$48.05_{6.64}$	$39.62_{3.14}$	$45.86_{1.70}$	$44.59_{2.10}$	$46.13_{5.19}$	$51.14_{3.87}$	$31.92_{3.63}$	$48.12_{2.37}$	$44.37_{0.99}$	$39.26_{2.25}$	$41.05_{1.38}$
Gemini 2.5 Flash Lite	$25.92_{1.82}$	$43.54_{4.48}$	$17.97_{1.92}$	$38.84_{3.82}$	$41.73_{1.47}$	$38.30_{3.10}$	$30.67_{0.88}$	$28.82_{2.38}$	$45.62_{7.49}$	$18.82_{0.68}$	$34.10_{3.78}$	$27.16_{0.32}$	$32.63_{2.66}$	$30.29_{0.42}$
GPT 5	$59.95_{2.01}$	$62.61_{2.59}$	$32.34_{2.08}$	$58.41_{1.63}$	$36.31_{1.60}$	$52.85_{4.72}$	$46.93_{2.83}$	$55.96_{3.14}$	$54.70_{4.54}$	$33.80_{2.20}$	$54.97_{2.43}$	$53.42_{1.23}$	$55.07_{1.24}$	$48.01_{0.32}$
GPT 5 Mini	$49.59_{3.74}$	$60.45_{2.40}$	$29.22_{1.71}$	$50.19_{5.53}$	$52.49_{0.44}$	$51.68_{1.47}$	$50.28_{4.75}$	$44.33_{4.96}$	$58.19_{3.94}$	$25.54_{2.20}$	$49.22_{3.11}$	$41.17_{2.73}$	$57.02_{0.53}$	$45.21_{1.21}$
GPT 5 Nano	$22.99_{2.64}$	$45.98_{1.02}$	$10.46_{1.71}$	$24.81_{0.65}$	$11.47_{1.21}$	$26.59_{7.45}$	$21.49_{1.41}$	$26.42_{3.26}$	$23.56_{6.12}$	$12.81_{0.67}$	$26.17_{1.63}$	$25.27_{1.60}$	$32.84_{4.80}$	$21.22_{0.46}$
Grok 4	$39.64_{1.89}$	$36.96_{1.16}$	$29.00_{1.49}$	$44.44_{2.79}$	$40.70_{1.13}$	$47.63_{1.86}$	$40.57_{1.60}$	$36.73_{1.65}$	$22.09_{2.86}$	$24.77_{1.78}$	$50.43_{4.90}$	$30.29_{1.28}$	$39.02_{0.20}$	$36.08_{0.53}$
Open-source Models														
Mistral/Pixtral Family														
Mistral Medium 3.1	$24.77_{2.76}$	$37.01_{5.96}$	$16.01_{1.49}$	$28.48_{2.48}$	$33.70_{1.23}$	$34.14_{1.20}$	$24.41_{1.06}$	$25.22_{2.51}$	$38.99_{3.41}$	$11.46_{2.32}$	$25.46_{2.65}$	$19.62_{2.62}$	$31.09_{2.35}$	$24.86_{0.98}$
Pixtral Large	$19.09_{1.82}$	$35.09_{2.33}$	$11.33_{1.33}$	$24.32_{3.36}$	$27.40_{2.29}$	$23.41_{1.21}$	$24.16_{1.09}$	$19.01_{4.66}$	$19.89_{1.10}$	$11.54_{2.50}$	$21.93_{1.34}$	$18.08_{0.62}$	$22.64_{0.67}$	$20.10_{0.41}$
Mistral Small 24B	$15.38_{1.93}$	$25.07_{4.25}$	$7.00_{1.35}$	$22.29_{1.84}$	$20.47_{1.46}$	$21.53_{2.01}$	$13.07_{2.67}$	$15.34_{3.68}$	$18.57_{1.81}$	$7.76_{1.09}$	$13.84_{3.49}$	$10.61_{1.77}$	$16.36_{2.46}$	$14.43_{0.41}$
Pixtral 12B	$8.76_{0.77}$	$24.19_{3.58}$	$6.74_{0.94}$	$17.49_{0.53}$	$14.12_{0.11}$	$16.46_{2.65}$	$11.66_{2.40}$	$11.44_{1.34}$	$6.83_{2.27}$	$6.17_{0.35}$	$15.06_{2.39}$	$9.60_{0.46}$	$12.94_{2.80}$	$11.20_{0.02}$
Google Gemma Family														
Gemma 3 27B	$20.31_{1.18}$	$40.90_{1.49}$	$13.75_{1.55}$	$31.71_{3.21}$	$34.93_{1.52}$	$27.36_{6.28}$	$26.72_{1.12}$	$24.07_{2.01}$	$23.85_{2.74}$	$9.43_{1.30}$	$20.66_{2.62}$	$18.61_{0.40}$	$20.81_{2.15}$	$22.53_{0.28}$
Gemma 3 12B	$15.15_{0.69}$	$36.60_{1.32}$	$10.52_{1.44}$	$27.91_{1.30}$	$28.79_{1.39}$	$27.44_{3.60}$	$19.20_{1.27}$	$22.40_{1.47}$	$17.25_{2.89}$	$7.23_{1.12}$	$21.01_{1.65}$	$13.43_{1.47}$	$23.41_{0.13}$	$18.76_{0.63}$
Gemma 3 4B	$12.43_{1.63}$	$34.23_{1.08}$	$8.91_{0.96}$	$19.67_{4.37}$	$22.50_{0.12}$	$21.25_{1.33}$	$15.59_{0.87}$	$18.21_{1.21}$	$13.54_{2.63}$	$6.84_{1.10}$	$19.56_{2.12}$	$14.68_{1.08}$	$13.45_{0.88}$	$15.47_{0.78}$
AIDC-AI Ovis2 Family														
Ovis2-34B	$15.90_{1.35}$	$40.15_{2.16}$	$9.87_{0.77}$	$19.44_{0.45}$	$23.97_{0.56}$	$29.46_{1.47}$	$19.43_{0.58}$	$20.27_{3.31}$	$22.91_{2.46}$	$9.18_{1.41}$	$21.86_{2.89}$	$18.77_{1.37}$	$16.78_{0.26}$	$18.50_{0.03}$
Ovis2-16B	$11.20_{1.67}$	$38.98_{0.75}$	$8.08_{0.18}$	$21.58_{1.27}$	$24.68_{0.80}$	$23.94_{3.50}$	$21.20_{3.52}$	$14.83_{3.00}$	$24.32_{1.31}$	$8.72_{1.57}$	$20.21_{0.84}$	$16.47_{0.63}$	$16.12_{1.92}$	$17.18_{0.50}$
Ovis2-8B	$9.80_{1.30}$	$33.62_{1.54}$	$6.07_{0.30}$	$19.18_{3.28}$	$19.45_{1.85}$	$21.02_{1.98}$	$18.37_{1.83}$	$13.51_{1.33}$	$19.81_{5.29}$	$8.04_{0.53}$	$17.42_{3.08}$	$13.17_{0.35}$	$14.77_{1.80}$	$14.46_{0.37}$
Ovis2-4B	$6.76_{1.75}$	$23.66_{3.93}$	$6.00_{0.27}$	$15.89_{2.76}$	$16.16_{1.17}$	$17.05_{3.15}$	$16.43_{1.51}$	$10.68_{2.89}$	$13.16_{0.84}$	$7.17_{0.50}$	$17.65_{3.01}$	$14.26_{0.58}$	$8.31_{1.00}$	$12.18_{0.11}$
Ovis2-2B	$6.14_{0.22}$	$16.10_{1.01}$	$5.30_{0.83}$	$13.74_{2.34}$	$12.24_{1.70}$	$13.64_{4.43}$	$11.99_{1.14}$	$11.27_{2.01}$	$6.57_{1.32}$	$7.28_{0.64}$	$11.33_{2.19}$	$9.73_{0.56}$	$8.98_{3.88}$	$9.54_{0.22}$
Ovis2-1B	$4.83_{0.91}$	$12.62_{2.58}$	$4.74_{0.31}$	$8.07_{1.07}$	$7.52_{0.71}$	$5.95_{1.12}$	$8.03_{0.98}$	$8.11_{1.97}$	$6.57_{2.40}$	$5.05_{0.98}$	$8.10_{2.55}$	$6.80_{1.38}$	$4.43_{1.13}$	$6.52_{0.25}$
OpenGVLab InternVL3.5 Family														
InternVL3.5 38B	$14.94_{0.63}$	$30.95_{4.82}$	$9.09_{1.57}$	$24.85_{1.52}$	$28.79_{0.27}$	$20.90_{4.44}$	$19.25_{1.90}$	$18.40_{0.19}$	$24.54_{2.47}$	$8.53_{0.17}$	$21.10_{0.84}$	$16.41_{1.98}$	$14.76_{2.12}$	$18.01_{0.39}$
InternVL3.5 14B	$15.50_{2.05}$	$26.81_{4.46}$	$8.26_{1.12}$	$20.72_{0.96}$	$24.64_{1.18}$	$17.41_{3.95}$	$14.67_{1.98}$	$17.70_{3.09}$	$26.45_{1.63}$	$7.74_{0.53}$	$15.76_{1.58}$	$12.05_{1.07}$	$19.72_{3.13}$	$16.04_{0.37}$
InternVL3.5 8B	$10.22_{1.08}$	$23.11_{3.12}$	$7.14_{0.57}$	$20.44_{1.87}$	$20.14_{1.89}$	$16.16_{3.35}$	$11.27_{1.56}$	$11.99_{2.92}$	$22.96_{2.08}$	$5.29_{0.79}$	$12.68_{1.73}$	$12.57_{0.25}$	$13.01_{1.22}$	$13.16_{0.82}$
InternVL3.5 4B	$7.70_{1.15}$	$23.33_{0.30}$	$7.72_{0.52}$	$19.71_{1.60}$	$23.20_{1.24}$	$18.84_{0.42}$	$15.11_{1.52}$	$14.98_{2.03}$	$25.72_{2.64}$	$6.48_{0.96}$	$13.83_{1.36}$	$11.78_{1.37}$	$14.90_{2.25}$	$14.09_{0.28}$
InternVL3.5 2B	$5.32_{0.25}$	$20.86_{4.13}$	$5.24_{0.34}$	$15.50_{1.86}$	$16.05_{0.87}$	$12.69_{2.87}$	$8.94_{1.54}$	$7.69_{1.60}$	$14.18_{1.71}$	$5.63_{1.26}$	$10.14_{3.14}$	$7.03_{0.51}$	$9.07_{2.28}$	$9.48_{0.49}$
InternVL3.5 1B	$3.21_{0.43}$	$7.94_{2.99}$	$3.39_{0.09}$	$10.32_{0.07}$	$9.12_{0.32}$	$5.74_{0.58}$	$3.29_{1.02}$	$7.79_{1.30}$	$10.22_{1.53}$	$3.24_{0.57}$	$7.31_{1.05}$	$2.93_{0.74}$	$5.64_{0.43}$	$5.43_{0.13}$
Qwen2.5-VL Family														
Qwen2.5 VL 72B	$16.53_{1.36}$	$31.30_{1.38}$	$11.80_{2.24}$	$25.55_{1.06}$	$28.46_{2.62}$	$23.55_{1.42}$	$19.72_{0.38}$	$25.86_{3.14}$	$32.32_{7.22}$	$9.97_{0.45}$	$21.02_{2.62}$	$19.36_{0.79}$	$25.31_{1.59}$	$20.58_{0.80}$
Qwen2.5 VL 7B	$10.33_{0.70}$	$21.04_{4.51}$	$5.95_{1.26}$	$18.96_{1.05}$	$20.49_{3.89}$	$18.50_{3.79}$	$13.70_{0.92}$	$17.00_{4.00}$	$13.26_{4.07}$	$6.71_{0.28}$	$14.06_{1.66}$	$12.35_{0.74}$	$13.28_{2.86}$	$13.15_{0.86}$
Qwen2.5 VL 3B	$6.08_{2.15}$	$18.49_{3.90}$	$2.82_{0.44}$	$12.76_{1.17}$	$11.54_{1.70}$	$13.76_{2.51}$	$9.22_{0.16}$	$6.89_{1.47}$	$10.14_{0.98}$	$4.88_{0.18}$	$10.31_{3.46}$	$7.85_{0.38}$	$6.54_{0.84}$	$8.20_{0.36}$
Qwen3-VL Family														
Qwen3-VL-235B-A22B-Instruct	$37.75_{2.29}$	$54.44_{3.96}$	$23.28_{1.93}$	$43.16_{3.45}$	$51.51_{1.76}$	$47.42_{4.00}$	$39.14_{2.65}$	$40.98_{4.03}$	$54.31_{4.08}$	$22.75_{2.42}$	$36.33_{2.92}$	$37.44_{1.74}$	$40.10_{3.23}$	$38.41_{0.76}$
Qwen3-VL-235B-A22B-Thinking	$34.19_{2.38}$	$52.12_{4.01}$	$23.97_{1.87}$	$47.30_{3.12}$	$49.19_{1.91}$	$38.37_{3.97}$	$30.02_{2.49}$	$34.18_{3.68}$	$56.51_{3.95}$	$20.29_{2.32}$	$34.12_{2.80}$	$33.04_{1.70}$	$34.87_{3.16}$	$35.47_{0.75}$
Qwen3-VL-32B-Instruct	$36.74_{2.17}$	$56.30_{3.78}$	$18.13_{1.62}$	$41.29_{3.18}$	$51.39_{1.86}$	$41.73_{3.76}$	$34.28_{2.47}$	$43.38_{3.72}$	$60.92_{3.77}$	$19.67_{1.87}$	$36.02_{3.15}$	$34.43_{1.66}$	$32.25_{3.06}$	$36.08_{0.73}$
Qwen3-VL-32B-Thinking	$33.92_{2.36}$	$52.39_{4.28}$	$19.76_{1.72}$	$38.21_{2.96}$	$51.94_{1.75}$	$35.66_{3.53}$	$29.38_{2.40}$	$38.16_{3.86}$	$64.57_{3.65}$	$19.19_{2.15}$	$35.57_{3.14}$	$35.04_{1.72}$	$37.59_{3.04}$	$35.49_{0.74}$
Qwen3-VL-30B-A3B-Thinking	$36.19_{2.23}$	$56.38_{3.50}$	$18.06_{1.73}$	$38.44_{3.21}$	$49.92_{1.87}$	$38.48_{3.81}$	$32.34_{2.65}$	$37.99_{3.87}$	$68.69_{3.71}$	$17.81_{2.13}$	$37.40_{2.63}$	$35.46_{1.66}$	$29.95_{3.04}$	$35.41_{0.74}$
Qwen3-VL-30B-A3B-Instruct	$31.13_{2.26}$	$54.71_{3.46}$	$18.86_{1.65}$	$42.02_{3.29}$	$40.40_{1.66}$	$34.18_{3.53}$	$31.94_{2.75}$	$36.53_{3.80}$	$51.38_{3.94}$	$15.12_{1.66}$	$30.22_{2.54}$	$25.17_{1.49}$	$29.65_{2.92}$	$30.92_{0.70}$
Qwen3-VL-8B-Thinking	$28.27_{2.12}$	$49.55_{3.61}$	$11.21_{1.17}$	$33.92_{2.70}$	$42.07_{1.78}$	$29.73_{3.69}$	$26.70_{2.37}$	$32.53_{3.87}$	$47.83_{3.91}$	$14.10_{1.70}$	$24.90_{2.30}$	$28.75_{1.59}$	$24.20_{2.80}$	$28.01_{0.67}$
Qwen3-VL-8B-Instruct	$25.31_{2.07}$	$45.65_{3.88}$	$13.61_{1.52}$	$29.97_{2.99}$	$34.85_{1.70}$	$27.46_{2.95}$	$25.27_{2.49}$	$27.98_{3.47}$	$35.94_{3.41}$	$10.66_{1.43}$	$24.41_{2.60}$	$20.40_{1.31}$	$25.07_{2.77}$	$24.51_{0.64}$
Qwen3-VL-4B-Thinking	$24.23_{2.08}$	$45.07_{3.71}$	$12.83_{1.35}$	$30.66_{2.86}$	$38.53_{1.75}$	$29.72_{3.39}$	$24.89_{2.35}$	$31.29_{3.31}$	$46.69_{4.21}$	$14.92_{1.89}$	$23.85_{2.42}$	$25.31_{1.39}$	$22.60_{2.54}$	$26.18_{0.65}$
Qwen3-VL-4B-Instruct	$20.23_{1.91}$	$21.00_{2.84}$	$9.94_{1.32}$	$31.04_{3.02}$	$23.02_{1.35}$	$21.62_{3.07}$	$18.57_{1.91}$	$21.35_{2.85}$	$35.00_{3.56}$	$7.69_{1.26}$	$18.40_{1.97}$	$11.23_{1.10}$	$20.83_{2.73}$	$18.05_{0.56}$
Qwen3-VL-2B-Thinking	$11.81_{1.33}$	$24.58_{3.13}$	$5.43_{0.97}$	$19.67_{2.24}$	$17.53_{1.39}$	$13.32_{2.18}$	$13.58_{1.47}$	$16.09_{2.89}$	$26.97_{3.27}$	$9.03_{1.38}$	$17.61_{2.22}$	$13.81_{1.06}$	$12.36_{1.88}$	$13.87_{0.47}$
Qwen3-VL-2B-Instruct	$11.13_{1.53}$	$19.71_{3.07}$	$5.28_{0.86}$	$19.82_{2.22}$	$17.43_{1.27}$	$12.21_{1.95}$	$9.17_{1.33}$	$14.72_{2.62}$	$12.77_{2.33}$	$6.43_{1.00}$	$12.83_{1.96}$	$5.88_{0.66}$	$14.04_{2.12}$	$11.15_{0.43}$
Other Open-source														
Skywork-R1V3-38B	$27.12_{0.74}$	$47.94_{2.92}$	$15.30_{1.63}$	$32.37_{2.44}$	$36.84_{0.69}$	$37.25_{1.80}$	$26.43_{2.63}$	$28.27_{1.95}$	$41.71_{4.53}$	$14.76_{1.96}$	$30.10_{2.73}$	$27.38_{0.26}$	$26.42_{0.26}$	$27.76_{0.58}$
Korean-specialized Models														
VARCO-VISION-2.0-14B	$11.90_{0.79}$	$34.76_{4.78}$	$7.94_{0.85}$	$17.83_{2.30}$	$22.03_{2.71}$	$23.46_{3.16}$	$21.89_{1.09}$	$14.05_{2.80}$	$12.68_{1.90}$	$7.80_{2.64}$	$18.84_{1.75}$	$14.97_{0.57}$	$13.31_{1.31}$	$15.55_{0.50}$
HyperCLOVA-3B	$8.42_{0.98}$	$29.74_{2.98}$	$6.33_{0.49}$	$15.17_{1.40}$	$18.54_{0.41}$	$15.80_{2.14}$	$13.38_{0.67}$	$13.43_{3.83}$	$9.86_{2.44}$	$6.16_{0.70}$	$14.20_{1.75}$	$16.21_{0.93}$	$9.53_{1.86}$	$12.66_{0.18}$
VARCO-VISION-2.0-1.7B	$8.09_{1.21}$	$21.34_{1.50}$	$5.95_{2.38}$	$16.07_{0.96}$	$17.79_{0.63}$	$16.22_{1.16}$	$12.70_{0.32}$	$12.88_{1.08}$	$12.54_{5.35}$	$8.11_{0.68}$	$12.81_{1.01}$	$12.13_{0.72}$	$10.46_{3.57}$	$11.87_{0.46}$

Table 15: Complete performance across all 13 categories for all evaluated models (scores in %). All scores are reported as mean$_{SE}$, where SE is the standard error over 3 independent runs (n=3).



(a) Performance scaling with model size. Accuracy rises up to ∼10B parameters but improves more slowly thereafter.



(b) Domain-level results. Health/Medical yields the highest accuracy, whereas Entertainment/Gaming remains the most challenging.

Figure 9: Scaling and domain-level performance on HAERAE-Vision.

E Additional Results & Analysis

E.1 Full Results

Table 15 reports the full category-wise results for the 45 evaluated models; we will continuously update the leaderboard with newly released models.

E.2 Performance by Model Scale

Grouping models by size tiers (Small $\leq$4B, Medium 8–14B, Large $\geq$30B) reveals a clear scaling trend: performance improves with size. Large models reach a mean score of 0.3009 (95% CI [0.2974, 0.3046]), more than double Medium (0.1460) and triple Small (0.0854). All pairwise differences are significant (permutation $p \approx 0.001$) with large effect sizes (Large–Small $\Delta = +0.2155$, $d \approx 0.78$), confirming that scaling reliably enhances multimodal reasoning.

However, gains become less pronounced beyond about 10B parameters. Accuracy still rises but with smaller marginal improvements (Figure 9a), indicating that scale alone cannot close the gap. Further progress likely depends on advances in reasoning and cultural grounding.

At the family level, commercial systems (Gemini, GPT, Sonar) consistently outperform open-weight models (e.g., InternVL3), with effect sizes around $d = 0.7$–1.2 (e.g., Gemini-2.5-Pro vs InternVL3 $\Delta \approx 0.49$, $d \approx 1.21$). Thus, both scaling and architectural or cultural factors jointly drive performance.

E.3 Performance by Domain

Performance varies widely across the 13 domains (global mean = 0.1987, range 0.1179–0.332). Health/Medical achieves the highest checklist satisfaction (0.332), followed by Science (0.250), while Entertainment/Arts (0.118) and Gaming (0.119) remain the most challenging.

Within all domains, large models ($\geq$30B) consistently outperform small models ($\leq$4B) (permutation $p < 0.05$), with the largest gains in Health/Medical ($\Delta = +0.189$) and Mathematics ($\Delta = +0.163$). Even in Gaming and Entertainment, scale effects remain positive though absolute performance stays low (Figure 9b).

E.4 Investigating Failure Modes

In Table 15, we observe that VARCO-VISION and HyperCLOVA X—two Korean-focused VLMs—underperform multilingual counterparts of similar scale. While the precise reasons remain unclear due to the closed nature of these models and limited information about their training, we propose two possible explanations:

(A) **Training Data Coverage.** Current benchmarks that capture progress on culturally grounded,

information-deficient queries are scarce. Model developers may not have explicitly emphasized such aspects in their training data, leading to weaker performance on this type of evaluation.

(B) **Pretraining Scale and Robustness.** Robustness to imperfect or fragmented user queries may emerge from exposure to large-scale, diverse pretraining corpora. Larger multilingual models are more likely to encounter noisy, colloquial, or partially specified inputs, thereby preparing them better for benchmarks of this kind.

E.5 Visual-Text Grounding Error Analysis

Table 6 shows that visual-text grounding (VTG) errors increase from 5.2% to 16.6% after explicitation. To understand whether this reflects newly introduced errors or pre-existing failures being reclassified, we tracked individual (question, model) pairs across conditions.

Error Tracking. Of the 461 VTG errors in the explicitated condition, 401 (87.0%) were *newly surfaced*—cases that were already errors under original queries but classified under different failure categories. Only 60 (13.0%) were *persistent* VTG errors present in both conditions. Additionally, 104 VTG errors from the original condition were resolved by explicitation.

Source of Newly Surfaced VTG Errors. The 401 newly surfaced cases were previously classified under the following failure categories in the original condition (multi-label; percentages sum to >100%):

Original Failure Category	%
Lack of explicitness	70.8
Object recognition	55.6
Procedural reasoning	29.2
Cultural concept mismatch	21.4

Table 16: Original failure categories of newly surfaced VTG errors. Most were previously masked by lack-of-explicitness failures.

Interpretation. This supports an *error unmasking* interpretation: under-specified queries produce vague responses that fail for surface-level reasons. Once explicitation removes this ambiguity, models are forced to engage with specific visual regions, exposing deeper grounding failures previously masked by the dominant lack-of-explicitness error mode. Supporting this, VTG error severity remains virtually identical across conditions (severe: 84.8% $\rightarrow$ 83.7%), indicating that explicitation reveals pre-existing failures rather than creating new ones.

Example. In one case (idx=22), the model's response to the original under-specified query was vague and non-committal, merely identifying the character's reading while omitting details—annotated as "lack of explicitness." When given the explicitated query, the model attempted a specific answer but misread the character 惹 as 芯—annotated as "visual-text grounding." The same question produced different error types because explicitation forced the model to engage with the specific visual region, shifting the failure from surface-level vagueness to a concrete grounding error.

F Error Annotation Methodology

F.1 Annotation Setup

We used Claude 3.5 Sonnet as the LLM judge for error annotation, accessed via OpenRouter API with `temperature=0.0` and `max_tokens=2048`. For each error case (model response with score < 1.0), the judge was provided with the original question, gold answer, checklist items, model response, and metadata (source, category, model name, score).

Root Cause (select one)	
language	Misunderstood Korean grammar, negation, particles, or expressions
cultural_knowledge	Lacked Korean-specific cultural/institutional knowledge
general_reasoning	Understood language and context but failed at reasoning
Failure Category (select 1–3)	
object_recognition	Fails to identify key objects in the image
spatial_reasoning	Misinterprets spatial relations
cultural_concept_mismatch	Misunderstands Korean-specific concepts or conventions
visual_text_grounding	Refers to the wrong region/entity relative to the question
procedural_reasoning	Fails to execute multi-step procedures
lack_of_explicitness	Misses explicit facts demanded by the checklist
other	None of the above fit
Severity	
minor	Almost correct; small missing detail
moderate	Mixed correctness; partially useful
severe	Largely incorrect or misleading

Table 17: Error annotation taxonomy.

F.2 Annotation Prompt

System prompt:
You are an impartial error analysis assistant for a Korean multimodal QA benchmark. Your job is to carefully inspect each example and classify the model's failure according to a predefined taxonomy. Follow the provided schema exactly. Think step by step, but ONLY return the final JSON object in your response. Do NOT include explanations outside the JSON. Be strict and consistent with the taxonomy definitions.
User prompt:
You are given one question-answering example from a Korean multimodal benchmark, together with a model's answer and a detailed checklist used for scoring. Your goal is to analyze WHY the model failed or was only partially correct.
Based on the question, gold answer, checklist, and model answer:
1. Decide the SINGLE most important root cause of failure: "language", "cultural_knowledge", or "general_reasoning"
2. Choose 1–3 failure_categories describing HOW the error manifests
3. Choose severity: "minor", "moderate", or "severe"
4. Provide analysis_comment: 2–3 sentences in Korean explaining why the answer is wrong or incomplete
[Output format]
Return ONLY a single JSON object:
{"root_cause": "...", "failure_categories": [...], "severity": "...", "analysis_comment": "..."}
[Metadata]
- source: {source} - category: {category} - question_idx: {question_idx} - model_name: {model} - model_score: {score}
[Question] {question}
[Gold answer] {answer}
[Checklist items] {checklist}
[Model answer] {model_response}