WHEN EXAMPLES FAIL TO TEACH: THE DEMONSTRATION GAP IN VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal in-context learning (ICL) can help or harm vision-language model predictions, complicating demonstration selection. To understand this difficulty, we quantify the structure of demonstration utility across 13 models, three multiple-choice visual question answering datasets, and 27 conditions through exhaustive one-shot query–demonstration evaluations, measuring changes in correct-answer support and prediction correctness. Models that solve many of the same questions at zero-shot often benefit from different demonstrations on questions they both get wrong, despite partial utility transfer. On zero-shot errors, demonstration main effects are small, while query and pairing effects dominate utility variation. Selecting demonstrations by their average rescue rates rarely yields consistent gains across held-out query splits. Measured content similarities show weak linear alignment with pair-specific rescue patterns. In two end-to-end settings, the evaluated retrievers do not consistently improve accuracy. As a simple example of using query characteristics, we combine a simple trained gate for likely errors with voting over separate one-shot predictions using random demonstrations. This procedure improves accuracy over zero-shot by 1.22 percentage points on A-OKVQA and 0.53 on AI2D, while breaking fewer correct answers than the retrievers. These findings suggest the potential value of studying multimodal ICL through the characteristics of queries themselves.

1 INTRODUCTION

Vision-language models (VLMs) have demonstrated strong capabilities in visual understanding and multimodal reasoning (Bai et al., 2025b). Their ability to interpret visual evidence and apply task knowledge raises a natural expectation: *the knowledge that enables a model to solve a task should also help it learn from relevant demonstrations*. In-context learning (ICL) offers a direct means of realizing this expectation by providing demonstrations at inference time, without updating model parameters. Existing work provides encouraging evidence: demonstration retrieval based on multimodal representations and learned demonstration selectors can improve performance across several tasks (Qin et al., 2024; Luo et al., 2024). These advances suggest that demonstrations can serve as useful experience for models that already possess substantial multimodal knowledge.

Despite this promise, current VLMs do not consistently benefit from demonstrations. Adding demonstrations can degrade performance, and the effects of context vary with the task and its presentation (Huang et al., 2025; Xu et al., 2025). Studies of visual context show that models can neglect demonstration images and rely heavily on textual patterns (Chen et al., 2025b), while analyses of reasoning demonstrations reveal limited use of the supplied reasoning information (Huang et al., 2025). Beyond these behavioral findings, mechanistic studies have examined the construction of task mappings and their transfer to query samples, identifying failures in applying information established within demonstrations (Wang & Li, 2026). Together, these studies have exposed concrete obstacles to learning from multimodal context.

Figure 1 makes a complementary puzzle visible. For both queries, LLaVA-1.5-7B is wrong at zero shot, yet each of three demonstrations that shares no subject, scene or question vocabulary with the query independently changes the answer to the correct option when used alone. None shares

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.

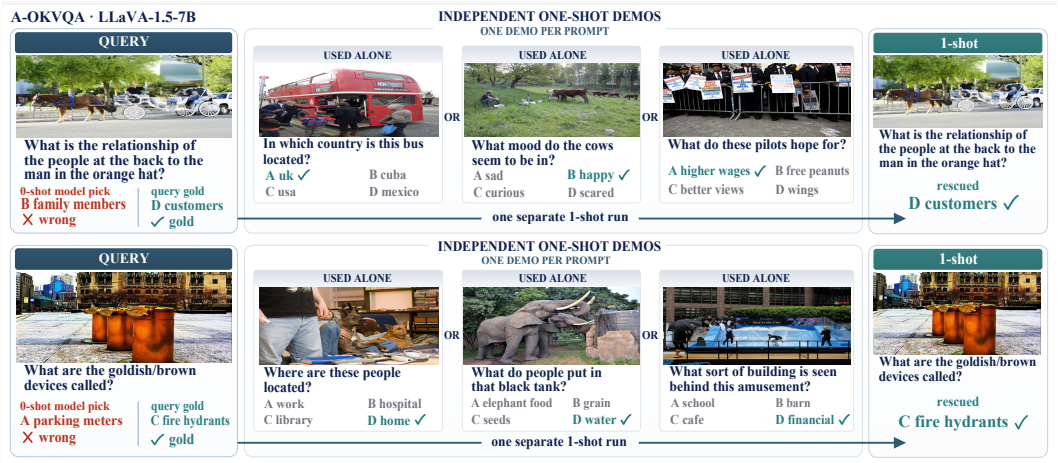

Figure 1: Seemingly unrelated demonstrations rescue the same zero-shot error. Each row shows a query that LLaVA-1.5-7B answers incorrectly at zero-shot and three unrelated demonstrations that each correct its prediction in a one-shot prompt. Checkmarks mark gold answers, and crosses mark zero-shot predictions.

the query's correct (*gold*) answer letter. These cases show that a demonstration can rescue a query without an obvious content match or answer-letter match, making its utility difficult to attribute to relevance alone.

These cases raise a direct question: how much do query, demonstration, and pairing effects contribute to variation in demonstration utility? We quantify these effects to understand the difficulties of demonstration selection. We test whether demonstration advantages persist across query splits and models, and whether answer symbols, text, or images reveal useful pair-specific structure.

We study this question in 27 conditions covering 13 models and three multiple-choice visual question answering datasets. For every condition, we randomly sample 500 queries and 500 candidate demonstrations with no image overlap between the two pools. We first compare zero-shot correctness and rescuing demonstrations between different sizes of the same architecture. We then split each continuous utility matrix into a query main effect, a demonstration main effect, and a pairing residual. For binary rescues, we separately examine how much residual variance is associated with answer letters, text similarity, and image similarity. These findings motivate a complementary, query-centric view of demonstration use. For example, rather than asking only which demonstration should be selected, we can also ask whether a query should be reconsidered using demonstrations at all. As a deliberately simple proof of concept, we use a gate to identify likely zero-shot errors and arbitrate over random one-shot demonstrations.

Our main findings are:

- **Shared task performance does not imply shared demonstration utility.** Related models share many zero-shot successes but overlap little in the demonstrations rescuing their shared errors. Continuous utility shows modest positive transfer across models, but models still differ substantially in how they benefit from the same query–demonstration pairs.
- **Query and pairing effects dominate utility variation.** Query effects are largest in 24 of 27 margin-based analyses on zero-shot errors; demonstration main effects only explain 0.28–8.62% of variance. Selecting demonstrations by their average rescue rates yields held-out gains whose reported split intervals lie entirely above zero in only three of 24 conditions.
- **Measured content similarity weakly tracks rescue patterns.** BM25 text and CLIP image similarity each capture less than 0.45% of binary rescue residual variance through linear alignment. Answer-letter associations can be substantial, but pairing variation persists in three controls that replace letters with option text.

- **Query characteristics can guide when demonstrations should be used.** In two settings, the evaluated retrievers do not consistently improve zero-shot accuracy. Gate and arbitration over random demonstrations improves it by 1.22 percentage points on A-OKVQA and 0.53 on AI2D, while breaking fewer correct answers than the retrievers.

These results identify limits of selection based on demonstration quality or the measured similarities, and suggest the potential value of studying multimodal ICL through the characteristics of queries themselves.

2 Experimental framework

We study how seeing a solved visual question changes a model's answer to another question. Each query contains an image, a question, and four answer options. A demonstration contains a separate image and question together with its correct answer. We first evaluate the query without a demonstration (zero shot), then place one demonstration before the same query and evaluate it again (one shot). Comparing the two predictions lets us observe whether that demonstration corrects an error, disrupts a correct answer, or leaves the answer unchanged.

Models and datasets. We use 13 vision–language models from LLaVA-1.5 (Liu et al., 2024), Qwen3-VL (Bai et al., 2025a), Qwen3.5 (Qwen Team, 2026a), Qwen3.8 (Qwen Team, 2026b), Gemma-3-IT (Gemma Team, 2025), Gemma-4 (Gemma Team, 2026), and Chameleon (Chameleon Team, 2024). The datasets are A-OKVQA (Schwenk et al., 2022), AI2D (Kembhavi et al., 2016) and PMC-VQA (Zhang et al., 2023).

Prompts and model predictions. In the main experiments, both queries and demonstrations list four options labelled A–D, and the demonstration ends with its correct (*gold*) answer letter. For the query, we score the four letter tokens at the answer position and take the most probable letter as the prediction. Both instruction wordings, *baseline* and *explicit*, request an A–D answer and differ only in the wording of this request. We call this *answer-letter prompting*. In the option-text control described in Sec. 4.3, answers are presented as option text without A–D labels. A *condition* specifies a model, a dataset, and a prompt configuration, including its instruction wording and answer format. The prespecified eligibility filter retains 27 answer-letter conditions.

Evaluating candidate demonstrations. For a given query, we repeat the one-shot evaluation with each candidate demonstration separately. This reveals which demonstrations change the answer to that question. Reusing each candidate across queries also reveals whether its benefits extend to other questions. Most comparisons in Sections 3 and 4 use 500 queries and 500 candidates per dataset, yielding $250{,}000$ separate one-shot evaluations per condition. Within these comparisons, models evaluated on the same dataset receive the same queries and candidates, with no images shared between the query and candidate pools. We arrange the observations in matrices whose rows are queries and whose columns are demonstrations. Section 5 uses a separate end-to-end evaluation protocol.

Rescues and breaks. The most direct benefit of a demonstration is to correct an initially wrong answer, which we call a *rescue*. Conversely, changing a correct zero-shot answer to an incorrect one is a *break*. Restricting the rows to the N_0 queries answered incorrectly at zero shot gives a binary rescue matrix: $X_{ij} \in \{0, 1\}$ records whether demonstration j changes the answer to query i from wrong to right. Append. A provides prompt templates, condition-selection criteria, cohort construction, and scoring details.

3 Task competence and demonstration utility

A demonstration that helps one model answer a question is a natural candidate for helping a related model answer the same question. We examine this possibility using different sizes of the same model family. These models may already solve many of the same questions without demonstrations, but it remains unclear whether they benefit from the same demonstrations when both initially get a question wrong. We therefore compare which questions the models solve on their own with which demonstrations correct their shared errors. This tests whether agreement in solving the task extends to the choice of rescuing demonstrations.

3.1 SHARED TASK PERFORMANCE AND RESCUING DEMONSTRATIONS

Experimental setup and metrics. We compare four model-size pairs sharing an architecture and training recipe, using matched queries and candidates. *Correct-answer containment* is the fraction of the smaller model's zero-shot successes also answered correctly by the larger model. On shared zero-shot errors, *rescue-set overlap* is the Jaccard overlap (i.e., intersection over union) of their rescuing demonstration sets, averaged over queries. We exclude candidates sharing the query's gold letter to limit answer-letter copying. Containment is compared against independent correctness, and overlap against independent sets of matched sizes. Their different denominators require separate chance references rather than a direct comparison of magnitudes. Append. B.1 gives the definitions and baselines.

Results and analysis. Models with the same architecture at different scales share task competence far more than demonstration utility. As shown in Tab. 1, containment is 79.9–97.7%, while rescue-set overlap on shared errors is only 3.9–7.1%. Across these four pairs, shared rescues occur at a median of 1.14 times the chance expectation, indicating only modest enrichment, as reported in Tab. 9 of Append. B.1. Binary rescue counts capture only wrong-to-right changes. The same demonstration may increase both models' support for the correct answer while changing the final prediction of only one. We therefore next examine continuous utility.

Table 1: Correct-answer containment and rescue-set Jaccard overlap for four model pairs with a shared architecture. Sample sizes and chance levels are reported in Tabs. 8–9 of Append. B.1.

Dataset	Model pair	Containment (%)	Rescue overlap (%)
AI2D	Chameleon 7B/30B	79.9	3.9
	LLaVA-1.5 7B/13B	88.9	6.3
A-OKVQA	Qwen3-VL 8B/32B	97.7	5.2
	Gemma-3-IT 4B/12B	89.8	7.1

3.2 CROSS-MODEL TRANSFER OF CONTINUOUS UTILITY

Experimental setup and metrics. The same demonstration can increase the gold answer's probability in both models while changing only one model's prediction from wrong to right. Binary rescue overlap misses such shared improvements. On all 500 queries and 500 candidates, including zero-shot successes, we measure the change in the gold answer's log-probability, denoted $M^{\log p}$. Positive values indicate increased support for the dataset-annotated correct answer. We remove query and demonstration main effects—the average utility associated with each query and each demonstration—to obtain the pair-specific residual R. Residual correlation ρ measures linear agreement across identical pairs in two models.

To test whether this agreement supports selection, a *source* model selects the 50 highest-residual candidates per query, excluding those with the query's gold letter. Each candidate is evaluated in a separate one-shot prompt. The *transfer gain* g is their mean residual utility on the *target* model minus that of all eligible candidates, averaged over queries. We report the *transfer factor* $\exp(g)$; one denotes no advantage over random selection, and values above one indicate positive transfer. The factor expresses an advantage in residual utility, not an accuracy ratio. Selection uses measured source-model utilities, making this a transfer diagnostic. Append. B.2 gives the metric definitions and decomposition formulas.

Results and analysis. Continuous utility shows modest positive transfer despite weak residual correlation across models. For the four pairs in Tab. 2, correlations range from 0.04 to 0.28, corresponding to at most eight percent shared variance under a linear fit. Nevertheless, demonstrations selected by one model have slightly higher residual utility in the other model than random demonstrations. The transfer factor is 1.05–1.11 in both directions, with 95% bootstrap intervals reported in Tab. 11 of Append. B.2. Across 128 ordered condition pairs on A-OKVQA and AI2D with matched datasets and instruction wordings, the median transfer factor in each group remains below 1.03, as reported in Tab. 10 of Append. B.2. The PMC-VQA group has a higher median factor of 1.140, and all its models belong to the Qwen family. These results suggest that demonstration utility has a small but nonzero component that transfers across models.

Table 2: Cross-model transfer of continuous utility within four model pairs that share an architecture. Correlation denotes residual correlation ρ for $M^{\log p}$. Arrows point from the model selecting demonstrations to the model being evaluated; these columns report transfer factors $\exp(g)$.

Dataset	Model pair	Correlation	Smaller → Larger	Larger → Smaller
AI2D	Chameleon 7B/30B	0.11	1.07	1.11
	LLaVA-1.5 7B/13B	0.28	1.09	1.06
A-OKVQA	Qwen3-VL 8B/32B	0.12	1.06	1.09
	Gemma-3-IT 4B/12B	0.04	1.08	1.05

Takeaway. Related models share substantial task performance, but this commonality is accompanied by much less agreement about which demonstrations rescue a query. Continuous analysis identifies transferable utility, while also showing that much of the pair-specific response remains model-dependent. We next examine what properties are associated with this utility within individual models.

4 Content and Answer Letters in Demonstration Use

The cross-model results leave open whether helpful demonstrations can be identified within an individual model. We distinguish reusable demonstration advantages from query-specific effects, then test their association with content similarity and answer letters.

4.1 Query, Demonstration, and Pairing Effects

Experimental setup and metrics. We decompose the $1{,}000 \times 1{,}000$ LLaVA-1.5-7B utility matrices on A-OKVQA and AI2D into query, demonstration, and pairing effects, following the decomposition in Sec. 3.2. The gold answer's probability may increase without increasing its advantage over competing answers. We therefore also use M^{margin}, the change in its log-probability margin over the highest-scoring incorrect option. This option is identified separately for each prompt. Both utility measures are evaluated on all queries and on the zero-shot error subset. We report the shares of total utility variance due to query main effects (Query), demonstration main effects (Dem.), and pairing residuals (Resid.). They sum to 100%.

Results and analysis. In these two conditions, demonstration main effects explain at most 1.19% across the reported scales and subsets, as shown in Tab. 3. Query effects explain 42.9–55.9% on A-OKVQA and 68.8–73.2% on AI2D. Pairing residuals retain 43.5–56.9% and 26.4–30.9%, respectively. For queries answered incorrectly at zero shot, query effects explain the largest share of variance in M^{margin} in 24 of 27 conditions. Demonstration effects explain 0.28–8.62% of variance, as shown in Tab. 12 of Append. C.1. The large query effects coexist with substantial pairing effects. Append. C.4 gives a complementary per-query view over $5{,}000$ candidate demonstrations. We next test whether demonstrations differ in their average rescue rates and whether these differences generalize to held-out queries.

4.2 Demonstration Heterogeneity and Held-Out Selection

Experimental setup and metrics. To test for demonstration advantages, an exchangeable null independently reassigns each query's rescues across candidates. It preserves how easy each query is to rescue while treating which demonstrations provide those rescues as random. The dispersion ratio (Disp.) divides the observed standard deviation of demonstration rescue rates by its null counterpart; one matches the null spread, and larger values indicate greater heterogeneity. The variance ratio (VR) is the observed variance divided by its null counterpart. The Kolmogorov–Smirnov (KS) distance is the largest absolute gap between the observed and null cumulative distributions; zero means agreement. We also report τ, which measures excess variation in demonstration rescue rates in units of the null standard deviation. When $\tau < 1$, this excess variation is smaller than the noise expected from random reassignment. The held-out selection test asks whether a demonstration chosen on one set of queries remains useful on another. We select the highest-rescue demonstration

Table 3: Variance decomposition of continuous utility into query main effects (Query), demonstration main effects (Dem.), and pairing residuals (Resid.). The two utility scales are $M^{\log p}$ and M^{margin}. Each is evaluated on all queries and on zero-shot errors. Values are percentages of total variance in the corresponding matrix.

		A-OKVQA			AI2D		
Matrix	Subset	Query	Dem.	Resid.	Query	Dem.	Resid.
$\Delta \log p$	all queries	42.91	0.20	56.90	71.87	0.10	28.03
	zero-shot errors	55.86	0.66	43.48	73.22	0.36	26.43
Δ margin	all queries	54.77	1.19	44.04	68.82	0.29	30.89
	zero-shot errors	51.54	0.68	47.77	69.99	0.34	29.67

on half of the zero-shot errors and measure its *held-out rescue-rate gain* over the mean candidate on the other half. Gains are in percentage points (pp) on errors only. We report means and 2.5th–97.5th percentile intervals over 200 splits; the intervals describe how the gain varies across splits of the same fixed matrix. The letter-copying score measures how often query and demonstration gold letters match among rescues, adjusted for chance. Scores of at least 25% define stronger letter copying. Append. C.2 provides the formulas.

Results and analysis. Across all 27 conditions, median Disp. is 1.12 and median KS is 0.0426. The typical distribution of demonstration rescue rates is close to the exchangeable null. Conditions with stronger letter copying have higher median VR, KS, and τ (4.592, 0.2407, and 1.90), indicating larger departures from the null. In 19 of the 20 conditions with weaker letter copying, $\tau < 1$, i.e. the additional spread in demonstration rescue rates is smaller than the null noise. The sole exception is Chameleon-7B on AI2D with explicit wording. Fig. 2 in Append. C.2 compares the observed and null rescue-count distributions for these 20 conditions.

Only three of the 24 reported intervals for held-out rescue-rate gain lie entirely above zero, as shown in Tab. 13 of Append. C.2: Chameleon-30B on A-OKVQA ($+21.3$ pp, $[14.0, 27.7]$), and Chameleon-7B on AI2D with baseline ($+16.5$ pp, $[7.1, 22.5]$) and explicit wording ($+8.9$ pp, $[3.0, 13.9]$). The two baseline Chameleon conditions have letter-copying scores above 25%; Chameleon-7B on AI2D with explicit wording falls below that threshold. As shown in Tab. 4, Disp. above one does not ensure gains that are robust across splits: Chameleon-30B on AI2D and both displayed LLaVA conditions have rescue-rate gain intervals crossing zero. Stable globally good demonstrations therefore appear rare in these conditions, i.e. selection by average rescue rate usually does not yield gains that remain positive across held-out splits.

4.3 Content similarity and answer-letter effects

Experimental setup and metrics. We next ask whether content similarity or answer letters help explain which demonstrations rescue a query. After pruning constant rows and columns, we double-centre the binary rescue matrix X to obtain R^X, distinct from the continuous utility residual R. We measure linear alignment by projecting R^X onto double-centred BM25 question-text similarity (Robertson & Zaragoza, 2009), CLIP image similarity (Radford et al., 2021), or query–demonstration gold-letter groups, and report the captured residual variance. The letter-associated share measures all gold-letter combinations, not the fraction of outputs copying a demonstration's letter. The projections share a denominator, but letter grouping has nine effective degrees of freedom versus one per similarity measure. Each condition retains its own zero-shot errors. Append. C.3 gives the formulas and chance references.

Results and analysis. Across all 27 conditions, BM25 and CLIP each explain less than 0.45% of binary rescue residual variance, with medians of 0.009% and 0.022%, as shown in Tab. 15 of Append. C.3. These scores capture little of the pair-specific rescue pattern that a content retriever would need to rank. This limited alignment does not rule out other forms of content structure. Across all five representation spaces, the mean cosine similarity between two rescuing demonstrations is only slightly higher than between a rescuing and a non-rescuing demonstration. The median gap is at most 0.026, and the largest gap in any condition is 0.083, as reported in Tab. 16 of Append. C.3.

Table 4: Variation in demonstration rescue rates, held-out selection, and pairing alignment under baseline answer-letter prompting. Rescue-rate gain is the held-out advantage on zero-shot errors, in pp. Intervals span the 2.5th–97.5th percentiles over 200 splits; bold gains have intervals above zero. The final three columns give binary rescue residual variance shares: letter association (Letter assoc.) with query–demonstration gold-letter groups, and linear alignment with BM25 text and CLIP image similarity, as defined in Sec. 4.3.

Model	Disp.	KS	Rescue-rate gain (pp)	Split interval (pp)	Letter assoc. (%)	BM25 (%)	CLIP (%)
A-OKVQA							
Gemma-3-4B-IT	0.96	0.0191	-0.5	$[-6.3, 5.5]$	1.14	0.000	0.059
Gemma-3-12B-IT	0.98	0.0126	-0.5	$[-6.6, 5.9]$	0.26	0.000	0.001
Chameleon-30B	3.90	0.5119	$\mathbf{+21.3}$	$[14.0, 27.7]$	24.71	0.001	0.000
AI2D							
Gemma-3-4B-IT	1.12	0.0585	-1.0	$[-9.4, 6.0]$	0.43	0.071	0.104
Chameleon-7B	3.67	0.3279	$\mathbf{+16.5}$	$[7.1, 22.5]$	17.31	0.004	0.010
Chameleon-30B	1.20	0.1524	$+2.3$	$[-4.3, 7.7]$	30.08	0.000	0.020
LLaVA-1.5-7B	1.23	0.0475	$+2.4$	$[-2.5, 6.6]$	0.12	0.437	0.314
LLaVA-1.5-13B	1.37	0.1099	$+6.2$	$[-2.2, 12.0]$	0.06	0.266	0.299

Table 5: Comparison of demonstration utility under answer-letter and option-text prompting in three conditions. Paired values for Disp., letter copying, and rescue-rate gain correspond to answer-letter and option-text prompting, respectively. Split intervals and Resid. refer only to option-text prompting. Resid. is the percentage of variance in M^{margin} attributable to query–demonstration pairing. Each format is evaluated on its own zero-shot errors. The letter-copying score differs from the letter-associated residual variance share.

Condition	Disp.	Letter copying (%)	Rescue-rate gain (pp)	Split interval (pp)	Resid. (%)
Gemma-4-12B / A-OKVQA	$2.62 \to 1.37$	$55.0 \to 0.8$	$+6.0 \to +3.9$	$[-1.2, +8.4]$	38.44
Gemma-4-12B / AI2D	$2.15 \to 1.14$	$60.9 \to 0.0$	$+7.2 \to +1.5$	$[-7.2, +7.8]$	31.16
Qwen3.5-9B / PMC-VQA	$1.40 \to 1.14$	$30.8 \to 0.4$	$+4.4 \to +1.4$	$[-2.8, +5.3]$	33.45

These small similarity gaps suggest that content similarity offers limited guidance for selecting demonstrations that can rescue a query.

Answer-letter association can be larger. It exceeds 2% in ten conditions and reaches about 50% for Gemma-4-12B on both datasets. It also depends on wording: with explicit instructions, Chameleon-7B on AI2D falls from 17.31% to 2.19%, whereas Qwen3.5-9B on A-OKVQA rises from 4.78% to 16.38%. A large letter-associated residual variance share does not necessarily yield a held-out rescue-rate gain. In Tab. 4, Chameleon-30B on AI2D has a 30.08% share, but its selection interval crosses zero.

Option-text control. For three conditions with high letter-copying scores, we remove A–D labels from query options and provide demonstration answers as option text without labelled option lists. As shown in Tab. 5, the letter-copying score falls from 30.8–60.9% to 0–0.8%. Disp. also falls to 1.14–1.37, and all held-out rescue-rate gain intervals cross zero. Pairing residuals still explain 31.2–38.4% of variance. In these three comparisons, lower letter-copying scores coincide with smaller global demonstration differences. Pair-specific variation remains under option-text prompting, but its size alone does not show that it can be predicted. The prompt, scoring, and format-specific details are in Append. A and C.3.

Takeaway. Query main effects account for a large share of continuous utility variation, and pairing residuals account for much of the rest. Demonstration main effects are small. Stable globally useful demonstrations are nearly absent in held-out selection. The measured content similarities capture little of the pairing variation. We next examine one possible use of query characteristics in multimodal ICL through an end-to-end comparison with content retrieval.

5 PRACTICAL CONSEQUENCES FOR DEMONSTRATION USE

Adding a demonstration can rescue a zero-shot error, but it can also break an answer that was already correct. The preceding analyses show why selecting a demonstration may not resolve this tradeoff. Demonstration main effects are small, and BM25 and CLIP capture little of the query–demonstration pairing variation. Query main effects are much larger. As a simple example of using query characteristics, we use a gate to identify likely zero-shot errors and consult random demonstrations only for those queries. Arbitration changes an answer only when the one-shot predictions provide sufficient agreement. We compare this control with content retrievers using overall accuracy, rescues, and breaks.

5.1 SELECTION, QUERY GATING, AND ARBITRATION

Retrieval baselines. We compare RICES image (Alayrac et al., 2022) (image similarity), RICES sum (Baldassini et al., 2024) (image and question similarity), MMICES (Chen et al., 2025a) (image filtering followed by text reranking), and MSIER (Luo et al., 2024) (trained on answering-model utilities). The first three are training-free, requiring no task-specific fitting. Each retriever selects k demonstrations and places them together before the query in a single prompt. We evaluate $k = 1$ (one demonstration) and $k = 4$ (four demonstrations). Random-single accuracy averages 20 separate draws, each evaluated in a separate one-shot prompt.

Gate and arbitration. A linear *gate* estimates zero-shot error likelihood from query representations and confidence features. Its development operating point targets a fraction r of queries for reconsideration; the other answers are retained. For each gated query, we obtain 20 separate one-shot predictions with random demonstrations. Under *arbitration*, the most-voted dissenting answer replaces the zero-shot answer only if it receives at least m votes without a tie; otherwise, the original answer remains. Random-single uses the same 20 draws. Append. D.2 gives the formal definitions of the gate threshold and arbitration rule.

Table 6: A-OKVQA / Gemma-3-4B-IT. End-to-end accuracy, rescues (wrong to right), and breaks (right to wrong). Random accuracy average 20 one-shot draws. Rescue and break counts are rounded to integers.

Method	k	Acc. (%)	Δ (pp)	Rescued	Broken
Zero-shot	–	78.08	–	–	–
Random single	1	75.66	−2.41	51	79
RICES image	1	75.98	−2.10	56	80
	4	76.33	−1.75	58	78
RICES sum	1	75.37	−2.71	63	94
	4	75.55	−2.53	67	96
MMICES	1	76.07	−2.01	71	94
	4	75.11	−2.97	58	92
MSIER	1	76.07	−2.01	65	88
	4	75.98	−2.10	54	78
Gate + arbitration	–	**79.30**	**+1.22**	18	4

Evaluation protocol. We evaluate Gemma-3-4B-IT on A-OKVQA and LLaVA-1.5-13B on AI2D. All methods share the candidate pool within a condition; MSIER and the gate train on queries from the same training split. We select r and m on a separate selection set and score each evaluation split once. We report accuracy (Acc.), change from zero shot (Δ, in pp), and counts of rescues and breaks as defined in Sec. 2. Their difference determines the net accuracy change. Append. D.1 gives the training recipes, splits, parameter selection, trigger rates, inference costs, and ablations.

5.2 RESULTS AND ANALYSIS: GAINS WITH FEWER BROKEN ANSWERS

Accuracy and retrieval baselines. Content retrieval does not yield a consistent accuracy improvement across the two settings. Tab. 6 shows that all nine demonstration baselines reduce A-OKVQA accuracy by 1.75–2.97 percentage points. On AI2D (Tab. 7), the eight retrieval results range from -1.16 to $+0.47$ points, with MMICES at k=1 providing the largest gain. The random-single mean is within 0.67 points of the best retrieval result in either setting; single-draw standard deviations are 0.63 points on A-OKVQA and 0.39 on AI2D. The trained MSIER retriever follows the same mixed pattern: both configurations reduce A-OKVQA accuracy, and neither exceeds the best training-free retriever on AI2D.

Gate and arbitration. Gate and arbitration improves accuracy by 1.22 pp on A-OKVQA and 0.53 pp on AI2D. On A-OKVQA, every demonstration baseline breaks more answers than it rescues.

Gate and arbitration rescues 18 errors and breaks 4 initially correct answers. By comparison, the baseline with the most rescues corrects 71 errors but breaks 94 correct answers.

Table 7: AI2D / LLaVA-1.5-13B.

Method	k	Acc. (%)	Δ (pp)	Rescued	Broken
Zero-shot	–	68.69	–	–	–
Random single	1	68.75	+0.06	76	75
RICES image	1	68.32	−0.37	98	105
	4	68.32	−0.37	121	128
RICES sum	1	68.95	+0.26	136	131
	4	68.37	−0.32	148	154
MMICES	1	69.16	+0.47	151	142
	4	68.11	−0.58	154	165
MSIER	1	68.95	+0.26	98	93
	4	67.53	−1.16	103	125
Gate + arbitration	–	**69.22**	**+0.53**	38	28

On AI2D, our procedure rescues 38 errors and breaks 28 answers, compared with 151 rescues and 142 breaks for the best retrieval result.

The ablation also shows the role of the gate. Applying arbitration to every query changes accuracy by $+0.44$ pp on A-OKVQA and -1.48 pp on AI2D, compared with $+1.22$ and $+0.53$ pp when the gate is used, as reported in Append. D.1.

Takeaway. Across these two settings, a simple query-based control with random demonstrations modestly outperforms the evaluated content retrievers in accuracy. It changes fewer predictions and breaks fewer initially correct answers. This illustrates one possible use of query characteristics in multimodal ICL.

6 Related work

Evaluating multimodal in-context learning. Benchmarks put a model's zero-shot accuracy next to its accuracy with demonstrations, and they report that the gain depends on the task and on how the demonstration is written (Zong et al., 2025; Qin et al., 2024). Chen et al. (2025b) report that models can ignore the demonstration image and follow the text. Huang et al. (2025) report that models take little from a supplied chain of reasoning. Xu et al. (2025) study how the choice and the presentation of the demonstrations change the outcome. Wang & Li (2026) study the task mapping that a demonstration builds and the point at which it fails to reach the query.

Choosing the demonstration. Retrieval methods rank candidate demonstrations by their estimated relevance or utility for a given query. RICES ranks the candidates by image similarity (Alayrac et al., 2022). The image-plus-text variant combines image similarity with question-text similarity (Baldassini et al., 2024). MMICES filters by image similarity and then reranks by text similarity (Chen et al., 2025a). MSIER trains a retriever using demonstration scores provided by the answering model (Luo et al., 2024). Our analysis instead quantifies query, demonstration, and pairing effects to characterize the difficulties of demonstration selection, and suggests the potential value of studying multimodal ICL through the characteristics of queries themselves.

7 Conclusion

Across 13 models, three multiple-choice visual question answering datasets, and 27 conditions, we find that task competence does not reliably translate into effective multimodal ICL. Models that solve many of the same questions at zero-shot often benefit from different demonstrations on questions they both get wrong, despite partial utility transfer. Demonstration main effects are small, while query and pairing effects account for most utility variation. Stable globally good demonstrations are nearly absent: only three of 24 reported held-out intervals lie entirely above zero. The evaluated content-based matching methods provide little guidance for identifying useful query–demonstration pairs. In two end-to-end conditions, a simple query-based control with random demonstrations modestly outperforms the evaluated retrievers in accuracy while changing fewer predictions and breaking fewer initially correct answers. Together, these findings characterize the difficulties of demonstration selection and suggest the potential value of studying multimodal ICL through the characteristics of queries themselves.

AI USE STATEMENT

We used large language models to assist with translation and manuscript polishing, including refining grammar and wording and improving the clarity and readability of the text. We also used these tools to assist with code development for the experiments.

ETHICS STATEMENT

All models and datasets used in this study are publicly available. We did not create or collect any new datasets.

REPRODUCIBILITY STATEMENT

Append. A–D provide the experimental details needed to reproduce our results, including model and dataset information, data sampling and splits, prompt templates, scoring procedures, metric definitions, statistical analyses, and training and evaluation settings. We will publicly release our code after cleanup.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025b. URL https://arxiv.org/abs/2502.13923.

Folco Bertini Baldassini, Mustafa Shukor, Matthieu Cord, Laure Soulier, and Benjamin Piwowarski. What makes multimodal in-context learning work? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1539–1550, 2024.

Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

Shuo Chen, Zhen Han, Bailan He, Jianzhe Liu, Mark Buckley, Yao Qin, Philip Torr, Volker Tresp, and Jindong Gu. Can multimodal large language models truly perform multimodal in-context learning? In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2025a.

Shuo Chen, Jianzhe Liu, Zhen Han, Yan Xia, Daniel Cremers, Philip Torr, Volker Tresp, and Jindong Gu. True multimodal in-context learning needs attention to the visual context. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=n4JdyBGu6T.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026.

Chengyue Huang, Yuchen Zhu, Sichen Zhu, Jingyun Xiao, Moises Andrade, Shivang Chopra, and Zsolt Kira. Mimicking or reasoning: Rethinking multi-modal in-context learning in vision-language models. *arXiv preprint arXiv:2506.07936*, 2025. URL https://arxiv.org/abs/2506.07936.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European Conference on Computer Vision (ECCV)*, 2016.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Yang Luo, Zangwei Zheng, Zirui Zhu, and Yang You. How does the textual information affect the retrieval of multimodal in-context learning? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5321–5335. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.emnlp-main.305. URL https://aclanthology.org/2024.emnlp-main.305/.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, et al. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Libo Qin, Qiguang Chen, Hao Fei, Zhi Chen, Min Li, and Wanxiang Che. What factors affect multi-modal in-context learning? an in-depth exploration. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/deeb4d6bdb5860fd7faf321dd5486d25-Abstract-Conference.html.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026a. URL https://qwen.ai/blog?id=qwen3.5.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, August 2026b. URL https://qwen.ai/blog?id=qwen3.8.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, 2021.

Stephen Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389, 2009.

Dustin Schwenk, Apoorv Khandelwal, Christopher Clark, Kenneth Marino, and Roozbeh Mottaghi. A-OKVQA: A benchmark for visual question answering using world knowledge. In *European Conference on Computer Vision (ECCV)*, 2022.

Yu Wang and Sharon Li. Why multimodal in-context learning lags behind? unveiling the inner mechanisms and bottlenecks. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13670–13685. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.622. URL https://aclanthology.org/2026.acl-long.622/.

Shitao Xiao, Zheng Liu, Peitian Zhang, Niklas Muennighoff, Defu Lian, and Jian-Yun Nie. C-Pack: Packed resources for general Chinese embeddings. In *International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, 2024.

Nan Xu, Fei Wang, Sheng Zhang, Hoifung Poon, and Muhao Chen. From introspection to best practices: Principled analysis of demonstrations in multimodal in-context learning. 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)*, pp. 3299–3324. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.170. URL https://aclanthology.org/2025.naacl-long.170/.

Xiao Xu, Chenfei Wu, Shachar Rosenman, Vasudev Lal, Wanxiang Che, and Nan Duan. BridgeTower: Building bridges between encoders in vision-language representation learning. In *AAAI Conference on Artificial Intelligence*, 2023.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

Xiaoman Zhang, Chaoyi Wu, Ziheng Zhao, Weixiong Lin, Ya Zhang, Yanfeng Wang, and Weidi Xie. PMC-VQA: Visual instruction tuning for medical visual question answering. *arXiv preprint arXiv:2305.10415*, 2023.

Yongshuo Zong, Ondrej Bohdal, and Timothy Hospedales. VL-ICL Bench: The devil in the details of multimodal in-context learning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=cpGPPLLYYx.

A Experimental framework and matrix protocol

Models. The matrix analysis covers 13 vision–language models: LLaVA-1.5 at 7B and 13B, Qwen3-VL at 8B and 32B, Qwen3.5 at 9B and 27B, Qwen3.8 at 27B, Gemma-3-IT at 4B, 12B and 27B, Gemma-4 at 12B, and Chameleon at 7B and 30B.

Datasets. The matrix analysis uses A-OKVQA, AI2D and PMC-VQA, and every question is four-way multiple choice. Each cohort is drawn within one source split. On A-OKVQA and AI2D, the cohort comes from the training split. On AI2D we keep the questions that parse strictly, carry `abcLabel=false` and do not refer to an abstract label drawn inside the diagram. PMC-VQA has no split we can hold out, so its cohort comes out of the cleaned internal test split, which holds $1{,}999$ questions over $1{,}440$ images.

Query and candidate construction. Each dataset has one frozen cohort that all its conditions reuse. The A-OKVQA and AI2D cohorts take $1{,}000$ queries and $1{,}000$ candidates. The PMC-VQA cohort takes 500 of each, because its split does not carry enough images to draw $1{,}000$ of each without sharing an image between the two sides. Within a cohort every query carries its own image, every candidate carries its own image, and no image appears on both sides. The candidate pool is stratified by gold answer letter, which gives exactly 250 per letter on A-OKVQA and AI2D and exactly 125 per letter on PMC-VQA. The queries are not stratified, so their gold letters keep the distribution of the split. The 500×500 matrix takes the first 500 queries and the first 500 candidates of the same cohort. The draws use fixed seeds, 20260709 for the candidates and 20260710 for the queries on A-OKVQA and on PMC-VQA, and 20260715 and 20260716 on AI2D. The frozen cohort files are checked into the repository.

Prompts. The prompts for each answer format are given below. A checkpoint that ships a chat template receives the body as one user turn through that template. A checkpoint without one receives the same plain layout shown here. Template availability follows the model release and is not an experimental setting. The opening instruction and answer instruction are each one line in the actual prompt and may wrap here to fit the column.

Zero shot, answer letter, baseline instruction:

```
Answer the final multiple-choice visual question.
Each image token corresponds to the image immediately below it.

Final question:
<image>
Question: <query question>
Choices:
A. <option A>
B. <option B>
C. <option C>
D. <option D>
Answer with the option letter only.
```

One shot, answer letter, baseline instruction:

```
Use the solved examples to answer the final multiple-choice
visual question.
Each image token corresponds to the image immediately below it.

Example 1:
<image>
```

```
Question: <demonstration question>
Choices:
A. <option A>
B. <option B>
C. <option C>
D. <option D>
Answer: <demonstration gold letter>

Final question:
<image>
Question: <query question>
Choices:
A. <option A>
B. <option B>
C. <option C>
D. <option D>
Answer with the option letter only.
```

For both zero-shot and one-shot answer-letter prompts, the explicit instruction changes only the final line of the corresponding baseline template. It replaces `Answer with the option letter only.` with the following literal line:

```
Select the correct option and respond with exactly one
letter:  A, B, C, or D.
```

Zero shot, option text:

```
Answer the final multiple-choice visual question.
Each image token corresponds to the image immediately below it.

Final question:
<image>
Question: <query question>
Choices:
- <option A>
- <option B>
- <option C>
- <option D>
Answer with the option text only, copied exactly.
```

One shot, option text:

```
Use the solved examples to answer the final multiple-choice
visual question.
Each image token corresponds to the image immediately below it.

Example 1:
<image>
Question: <demonstration question>
Answer: <demonstration gold option text>

Final question:
<image>
Question: <query question>
Choices:
- <option A>
- <option B>
- <option C>
- <option D>
Answer with the option text only, copied exactly.
```

Scoring. We never let the model generate text and then parse it. Under answer-letter prompting, the answer is a single token. We read the conditional log-probabilities of the four letter tokens at the answer position and take the largest as the prediction. Scoring is deterministic, and a rerun reproduces a matrix byte for byte.

Under option-text prompting, we score the full text of each answer option. For each query–demonstration pair, we create four inputs: the same prompt followed by each of the four options. We score each option by how likely the model is to produce its full text as the answer. For an option containing two tokens, we read the probability of the first token after the prompt, and the probability of the second token after the prompt plus the first token. The option's score is the sum of the logarithms of these two probabilities. We apply the same rule to options of any length and select the option with the highest score. This procedure is called teacher-forced scoring.

We evaluate the four options together in one batch. Because their texts may have different lengths, we add padding so that all four inputs have the same length. Only the option tokens contribute to the score; prompt and padding tokens are excluded. To locate the option tokens correctly with either left or right padding, we use the attention mask to find the last non-padding token and count back by the option's token length. For the first few query–demonstration pairs in each run, we check that repeating a forward pass reproduces all four option scores exactly. We also evaluate four identical copies of one input together and check that all four receive exactly the same score. The run stops if either check fails.

Condition selection. A condition specifies a model, a dataset, and a prompt configuration, including instruction wording and answer format. An answer-letter condition enters the main analysis when the model's zero-shot accuracy on the 500 queries exceeds the chance ceiling by at least four points and when at least 50 zero-shot errors remain. The chance ceiling is the share of the most common gold letter among those 500 queries, which is 26.6% on A-OKVQA, 28.2% on AI2D and 37.8% on PMC-VQA. The first rule excludes conditions in which the model performs near chance. In these conditions, a wrong-to-right answer change may occur simply because the model switches to the correct letter by chance. The second rule removes the conditions where too few errors remain to measure anything.

Sample counts. A condition runs all $500 \times 500 = 250{,}000$ pairs as separate one-shot prompts. The two conditions behind the decomposition of Tab. 3 run the full $1{,}000 \times 1{,}000$ cohort, which is 10^6 pairs each. The binary rescue matrix retains only zero-shot errors, as defined in Append. B.1. Their number runs from 53 to 276 under baseline instruction wording. Each option-text run uses the zero-shot-error subset obtained under that answer format.

B TASK COMPETENCE AND CROSS-MODEL TRANSFER

The following notation is shared by the task-competence and within-model analyses. For the matrix analyses, $i \in \{1, \ldots, N\}$ indexes queries and $j \in \{1, \ldots, D\}$ indexes candidate demonstrations. The gold answer to query i is y_i. Its zero-shot prediction is z_i, and its prediction with demonstration j is z_{ij}. The demonstration's gold answer is y_j^{demo}. The symbol $p_{0,i}(a)$ denotes the forced-choice probability assigned to answer a at zero shot. The symbol $p_{1,ij}(a)$ denotes the corresponding probability after adding demonstration j.

B.1 SHARED TASK PERFORMANCE AND RESCUING DEMONSTRATIONS

Binary rescue. The rescue matrix is defined only on the N_0 queries answered incorrectly at zero shot:

$$X_{ij} = \mathbf{1}\{z_i \neq y_i \text{ and } z_{ij} = y_i\}.$$

Thus, $X_{ij} = 1$ means that demonstration j corrects query i. The rescue rate of demonstration j is $\hat{p}_j = N_0^{-1} \sum_i X_{ij}$. Larger values mean that the demonstration corrects more of the fixed zero-shot errors. This quantity does not count damage to queries that were already correct.

Correct-answer containment. Let C_s and C_l be the sets of queries answered correctly at zero shot by the smaller and larger models, respectively. We report

$$\text{Containment} = \frac{|C_s \cap C_l|}{|C_s|}, \qquad \text{Independence} = \frac{|C_l|}{N}.$$

Containment is one when every success of the smaller model is also a success of the larger model. The independence value is its reference under independent correctness. The miss ratio in Tab. 8 is

$$\text{Miss ratio} = \frac{|C_s \setminus C_l|}{|C_s|(1 - |C_l|/N)}.$$

A miss ratio of zero means that the larger model answers correctly every question that the smaller model answers correctly. A ratio of one means that the larger model's error rate on these questions equals its error rate across all N questions.

Rescue-set overlap. For a query that both models answer incorrectly at zero shot, let A_i and B_i be their rescue sets after excluding candidates with the query's gold letter. Let $a_i = |A_i|$, $b_i = |B_i|$, $c_i = |A_i \cap B_i|$, and let D_i be the number of eligible candidates. The query-level overlap is

$$J_i = \frac{c_i}{a_i + b_i - c_i}.$$

We average J_i over queries with a nonempty union. Its chance reference replaces the observed intersection by $a_i b_i / D_i$:

$$J_{0,i} = \frac{a_i b_i / D_i}{a_i + b_i - a_i b_i / D_i}.$$

The enrichment column in Tab. 9 uses pooled intersections,

$$\text{Enrichment} = \frac{\sum_i c_i}{\sum_i a_i b_i / D_i}.$$

An enrichment of one matches independently placed rescue sets of the observed sizes. Values above one indicate more shared rescues than this reference. The Jaccard overlap and containment have different denominators and should not be compared as if they were the same scale.

Which queries the two models get right. Tab. 8 gives every pair for which both models have a zero-shot run on the same cohort. The blocks separate pairs with at least 20 misses because ratios based on fewer misses are unstable. In the upper block, the miss ratio ranges from 0.32 to 0.59, and containment exceeds its independence reference throughout.

Table 8: Correct-answer containment between two models at zero shot. Containment is the percentage of the smaller model's correct answers that the larger model also answers correctly. The independence baseline equals the larger model's overall zero-shot accuracy. A miss is a query answered correctly by the smaller model but incorrectly by the larger model. Miss ratio is the actual miss count divided by the expected count under independence; a ratio of 0.32 means 32% of the expected misses.

Larger model	Smaller model	Dataset	Containment (%)	Independence baseline (%)	Misses (count)	Miss ratio
At least 20 misses						
Gemma-3-27B-IT	Gemma-3-12B-IT	A-OKVQA	94.9	83.8	20	0.32
Qwen3.5-27B	Qwen3.5-9B	PMC-VQA	89.3	66.6	33	0.32
Qwen3-VL-32B	Qwen3-VL-8B	PMC-VQA	86.6	59.6	35	0.33
Qwen3.8-27B	Qwen3.5-9B	A-OKVQA	95.1	86.2	21	0.35
Gemma-3-27B-IT	Gemma-3-4B-IT	A-OKVQA	94.1	83.8	22	0.36
LLaVA-1.5-13B	LLaVA-1.5-7B	AI2D	88.9	72.4	37	0.40
Gemma-3-12B-IT	Gemma-3-4B-IT	A-OKVQA	89.8	78.2	38	0.47
Chameleon-30B	Chameleon-7B	AI2D	79.9	66.2	45	0.59
Fewer than 20 misses						
LLaVA-1.5-13B	LLaVA-1.5-7B	A-OKVQA	99.4	97.0	3	0.22
Qwen3-VL-32B	Qwen3-VL-8B	AI2D	99.0	98.2	5	0.57
Gemma-3-27B-IT	Gemma-3-12B-IT	AI2D	98.7	95.4	6	0.28
Qwen3.5-27B	Qwen3.5-9B	AI2D	98.8	97.8	6	0.56
Qwen3.5-27B	Qwen3.5-9B	A-OKVQA	97.9	89.4	9	0.20
Gemma-3-27B-IT	Gemma-3-4B-IT	AI2D	97.7	95.4	10	0.50
Qwen3-VL-32B	Qwen3-VL-8B	A-OKVQA	97.7	90.2	10	0.24
Qwen3.8-27B	Qwen3.5-9B	AI2D	97.3	96.0	13	0.67
Gemma-3-12B-IT	Gemma-3-4B-IT	AI2D	95.7	92.2	19	0.56

Which demonstrations rescue both models. Tab. 9 expands the rescue-set overlap results in Tab. 1 with their chance baselines. Shared errors is the number of queries both models answer incorrectly at zero shot and is the sample size of each row. The Qwen3-VL row on AI2D is omitted from the main table because it has only four shared errors.

Table 9: Rescue-set overlap between the model pairs listed under Architecture on each Dataset. Shared errors counts queries that both models answer incorrectly at zero shot. For each such query, Overlap is the percentage of demonstrations that rescue both models among those that rescue at least one; we average over queries with at least one rescue. Chance is the corresponding reference percentage for independent rescue sets of the same sizes. Enrichment divides the total number of shared rescues across queries by its expected total under independence: 1.11 means 11% more shared rescues than expected. It is not Overlap divided by Chance. Demonstrations whose gold letter matches the query's are excluded from both rescue sets. The Qwen3-VL result on AI2D is based on only four shared errors and should be interpreted cautiously.

Architecture	Dataset	Overlap	Chance	Enrichment	Shared errors
Chameleon 7B/30B	AI2D	3.9	3.2	1.11	124
LLaVA-1.5 7B/13B	AI2D	6.3	1.6	1.17	101
Qwen3-VL 8B/32B	A-OKVQA	5.2	1.5	1.32	39
Qwen3-VL 8B/32B	AI2D	6.0	0.8	1.45	4
Qwen3-VL 8B/32B	PMC-VQA	6.6	4.6	1.15	167
Gemma-3-IT 4B/12B	A-OKVQA	7.1	6.0	1.03	71

B.2 Cross-model transfer of continuous utility

Continuous utility. We use two continuous matrices:

$$M_{ij}^{\log p} = \log p_{1,ij}(y_i) - \log p_{0,i}(y_i),$$

and

$$M_{ij}^{\text{margin}} = \left[\log p_{1,ij}(y_i) - \max_{a \neq y_i} \log p_{1,ij}(a)\right] - \left[\log p_{0,i}(y_i) - \max_{a \neq y_i} \log p_{0,i}(a)\right].$$

The first measures the change in support for the gold answer. It tells us how much more or less confident the model becomes in the correct answer after seeing a demonstration. The second measures the change in its lead over the strongest non-gold answer. It tells us whether the correct answer gains or loses ground against the model's most likely wrong answer, and by how much. The strongest non-gold answer is recomputed under each prompt and can therefore change after a demonstration is added. Zero means no change on the given scale. Positive values favor the gold answer, and negative values move against it.

Additive decomposition. For either continuous utility matrix M, let $\mu = \bar{M}_{\cdot\cdot}$, $u_i = \bar{M}_{i\cdot} - \mu$, and $\alpha_j = \bar{M}_{\cdot j} - \mu$. We write

$$M_{ij} = \mu + u_i + \alpha_j + R_{ij}, \qquad R_{ij} = M_{ij} - \bar{M}_{i\cdot} - \bar{M}_{\cdot j} + \mu.$$

Here u_i is the query main effect, α_j is the demonstration main effect, and R_{ij} is the query–demonstration pairing residual. A positive R_{ij} means that the pair works better than expected from its query and demonstration averages.

Let $\text{SS} = \sum_{ij}(M_{ij} - \mu)^2$. The reported variance shares are

$$\text{Query} = \frac{D\sum_i u_i^2}{\text{SS}}, \qquad \text{Dem.} = \frac{N\sum_j \alpha_j^2}{\text{SS}}, \qquad \text{Resid.} = \frac{\sum_{ij} R_{ij}^2}{\text{SS}}.$$

They sum to one before conversion to percentages. A larger share means that the corresponding component describes more variation in the fixed utility matrix.

Residual transfer. Let R^{s} and R^{t} be source and target residuals on matched query–demonstration pairs. For query i, $\mathcal{E}_i$ is the set of candidates whose gold letter differs from the query's, and T_i contains the 50 candidates in $\mathcal{E}_i$ with the highest source residual. We define

$$\rho = \text{corr}\big(R_{ij}^{\text{s}}, R_{ij}^{\text{t}}\big)$$

over identical eligible pairs, and

$$g = \frac{1}{N}\sum_i \left[\frac{1}{50}\sum_{j \in T_i} R_{ij}^{\text{t}} - \frac{1}{|\mathcal{E}_i|}\sum_{j \in \mathcal{E}_i} R_{ij}^{\text{t}}\right].$$

The first quantity measures linear agreement in pair-specific utility. The second measures the target residual advantage of source-selected candidates over uniform selection from eligible candidates. We report $\exp(g)$ so that one means no advantage, values above one mean positive transfer, and values below one mean negative transfer. This factor is not an accuracy ratio.

Continuous utility and transfer factors. Unlike the binary-rescue results in Tab. 9, the next two tables use $M^{\log p}$ on the full 500×500 matrices, including zero-shot successes.

Transfer across all condition pairs. Tab. 2 covers four within-architecture size pairs. Tab. 10 covers every directed pair among conditions that share a dataset and instruction wording, including pairs from different architectures. The directions $A \to B$ and $B \to A$ are distinct. Each row reports medians over its directed pairs, which total 140 across the five rows.

Table 10: Transfer over all 140 ordered pairs of the 27 conditions. Correlation and Factor denote residual correlation ρ and transfer factor $\exp(g)$. The four models of the PMC-VQA group all come from the Qwen family, so that group carries no pair from two different model families.

Dataset	Instruction	Pairs	Correlation	Factor
A-OKVQA	baseline	72	0.029	1.026
A-OKVQA	explicit	20	0.045	1.027
AI2D	baseline	30	0.015	1.013
AI2D	explicit	6	0.006	1.002
PMC-VQA	baseline	12	0.152	1.140

Bootstrap intervals for transfer factors. Transfer gain is the mean residual utility difference over 500 queries. To assess how this mean varies with the queries included, we use bootstrap resampling. In each round, we randomly select one query from the original 500 and record its utility difference. We make 500 draws in total, each time selecting from all 500 original queries, including those already drawn. A query may therefore appear several times, while others may not appear at all. We average the 500 recorded differences to obtain one value of g. Repeating the entire round $2{,}000$ times gives $2{,}000$ values of g. We exponentiate their 2.5th and 97.5th percentiles to obtain the 95% interval for the transfer factor $\exp(g)$. All rounds reuse the utility differences already computed for the two models on the same fixed cohort. Tab. 11 reports these intervals for both transfer directions of each model pair.

Table 11: Bootstrap intervals for transfer factors in both directions for the four model pairs in Tab. 2 and the Qwen3-VL pair on PMC-VQA. The Selected by column identifies the source model; the other model in the pair is the target. Factor denotes the transfer factor $\exp(g)$ on the target model.

Architecture	Dataset	Selected by	Factor	95% interval
Chameleon 7B/30B	AI2D	7B	1.069	$[1.059, 1.078]$
Chameleon 7B/30B	AI2D	30B	1.110	$[1.098, 1.123]$
LLaVA-1.5 7B/13B	AI2D	7B	1.086	$[1.070, 1.107]$
LLaVA-1.5 7B/13B	AI2D	13B	1.061	$[1.050, 1.072]$
Qwen3-VL 8B/32B	A-OKVQA	8B	1.059	$[1.039, 1.088]$
Qwen3-VL 8B/32B	A-OKVQA	32B	1.088	$[1.060, 1.118]$
Qwen3-VL 8B/32B	PMC-VQA	8B	1.230	$[1.179, 1.283]$
Qwen3-VL 8B/32B	PMC-VQA	32B	1.361	$[1.293, 1.444]$
Gemma-3-IT 4B/12B	A-OKVQA	4B	1.076	$[1.046, 1.112]$
Gemma-3-IT 4B/12B	A-OKVQA	12B	1.054	$[1.023, 1.086]$

C WITHIN-MODEL DEMONSTRATION UTILITY

C.1 QUERY, DEMONSTRATION, AND PAIRING EFFECTS

Continuous utility decomposition. Section 4.1 uses the utility and decomposition definitions in Append. B.2. Tab. 3 uses the full $1{,}000 \times 1{,}000$ LLaVA-1.5-7B cohorts. Both $M^{\log p}$ and M^{margin} are decomposed on all queries and on zero-shot errors. The 500×500 matrices used elsewhere take the first 500 queries and candidates from the same frozen cohorts.

Variance decomposition across conditions. Tab. 12 extends the continuous utility decomposition in Tab. 3 to all 27 conditions, using M^{margin} on zero-shot errors. The query main effect is largest in 24 of the 27 conditions, and the demonstration main effect accounts for 0.28–8.62%. Pairing residuals are largest for Gemma-4-12B on A-OKVQA and AI2D and for Qwen3.5-9B on PMC-VQA.

Table 12: Variance decomposition of M^{margin} on zero-shot errors across all 27 conditions. Query, Dem., and Resid. denote the variance shares of query main effects, demonstration main effects, and continuous utility pairing residuals, as in Tab. 3. All values are percentages of total matrix variance. All conditions use answer-letter prompting. Wording identifies the instruction variant.

Model	Wording	Query	Dem.	Resid.
A-OKVQA				
Gemma-3-4B-IT	Baseline	54.78	0.42	44.80
	Explicit	52.87	0.50	46.63
Gemma-3-12B-IT	Baseline	71.16	0.28	28.56
Gemma-3-27B-IT	Baseline	61.11	0.64	38.26
Gemma-4-12B	Baseline	36.66	5.38	57.96
Chameleon-30B	Baseline	62.62	8.62	28.76
LLaVA-1.5-7B	Explicit	55.88	1.27	42.85
Qwen3-VL-8B	Baseline	66.16	1.25	32.59
	Explicit	66.10	1.05	32.85
Qwen3-VL-32B	Explicit	75.85	0.72	23.43
Qwen3.5-9B	Baseline	57.18	1.33	41.49
	Explicit	53.48	1.53	44.99
Qwen3.5-27B	Baseline	78.06	0.80	21.14
Qwen3.8-27B	Baseline	80.63	0.60	18.78
AI2D				
Gemma-3-4B-IT	Baseline	67.42	1.12	31.46
	Explicit	61.71	0.87	37.42
Gemma-4-12B	Baseline	25.13	4.77	70.10
Chameleon-7B	Baseline	78.56	3.00	18.44
	Explicit	63.94	2.99	33.07
Chameleon-30B	Baseline	66.27	4.02	29.72
LLaVA-1.5-7B	Baseline	69.89	0.41	29.71
	Explicit	65.96	0.65	33.39
LLaVA-1.5-13B	Baseline	56.34	1.08	42.58
PMC-VQA				
Qwen3-VL-8B	Baseline	58.91	0.69	40.39
Qwen3-VL-32B	Baseline	57.45	0.79	41.76
Qwen3.5-9B	Baseline	43.90	1.81	54.29
Qwen3.5-27B	Baseline	65.40	1.21	33.39

C.2 DEMONSTRATION HETEROGENEITY AND HELD-OUT SELECTION

Exchangeable-demonstration null. For each query, the null preserves the observed rescue count $\sum_j X_{ij}$ and uniformly reassigns those rescues across the D candidates. It therefore preserves how easy each query is to rescue while treating demonstrations as exchangeable: each candidate has the same chance of being assigned a rescue for that query. Let $\bar{p}_i = D^{-1} \sum_j X_{ij}$ be the rescue rate of query i. The analytical null standard deviation of demonstration rescue rates is

$$\sigma_0 = \sqrt{N_0^{-2} \sum_i \bar{p}_i (1 - \bar{p}_i)}.$$

In the implementation, v_0 and s_0 are the mean variance and mean standard deviation, respectively, of demonstration rescue rates across null reassignments; s_{obs} is their observed standard deviation. We use 500 null reassignments to estimate v_0 and 9,999 to estimate s_0. We report

$$\text{VR} = \frac{s_{\text{obs}}^2}{v_0}, \qquad \text{Disp.} = \frac{s_{\text{obs}}}{s_0}, \qquad \tau = \sqrt{\max(\text{Disp.}^2 - 1, 0)}.$$

VR and Disp. compare the observed spread with null noise: values below, equal to, or above one indicate smaller, equal, or greater spread relative to null noise, respectively. The quantity τ expresses inferred excess spread in null-noise standard deviation units. In particular, $\tau < 1$ means that the excess spread is smaller than one standard deviation of null noise. A value of $\tau = 0$ means that the observed spread in demonstration rescue rates is equal to or smaller than the spread expected under the null model.

Distributional distance. Under the same null, the rescue count of a candidate follows the Poisson-binomial distribution induced by the query rescue rates $\bar{p}_i$. If F_{obs} and F_0 are the observed and null cumulative distributions of candidate rescue counts, then

$$\text{KS} = \max_k |F_{\text{obs}}(k) - F_0(k)|.$$

KS is zero when the two distributions agree. Larger values mean greater distributional deviation. It has no universal substantive threshold here and is reported as a distance rather than a rejection decision.

Ten-nearest-neighbour enrichment. This diagnostic asks whether high-rescue demonstrations are locally grouped in the continuous utility geometry. On zero-shot errors, let $R^{\log p} \approx U_r \Sigma_r V_r^\top$ be the rank-r approximation of the continuous $M^{\log p}$ pairing residual. Demonstration j has coordinate

$$\mathbf{d}_j = (V_r \Sigma_r^{1/2})_{j\cdot}.$$

The retained rank is the consecutive singular-value prefix above the rank-specific 95th percentile of a row-permutation null. The implementation retains at least one component. Let H be the demonstrations at or above the 90th percentile of binary rescue counts, and let $\text{NN}_{10}(j)$ be the ten Euclidean nearest neighbours of $\mathbf{d}_j$. The enrichment is

$$E_{10} = \frac{|H|^{-1} \sum_{j \in H} \frac{1}{10} \sum_{\ell \in \text{NN}_{10}(j)} \mathbf{1}\{\ell \in H\}}{(|H| - 1)/(D - 1)}.$$

If fewer than ten neighbours exist, all available neighbours are used and the same count replaces 10. The high-rescue threshold uses the higher empirical quantile. If ties would place every candidate in H, the stable top $\lceil 0.1D \rceil$ candidates are used. A value of one matches the random baseline. Values above one mean that high-rescue demonstrations have more high-rescue neighbours on average than expected by chance in the low-rank continuous residual space. For example, a value of 1.1 means that high-rescue demonstrations have 10% more high-rescue neighbours on average than expected by chance.

Held-out selection. For split t, divide the zero-shot errors into equal selection and evaluation halves A_t and B_t. Let $\hat{p}_{A_t j}$ and $\hat{p}_{B_t j}$ be the rescue rates of candidate j on the two halves. We select

$$j_t^* = \arg\max_j \hat{p}_{A_t j}$$

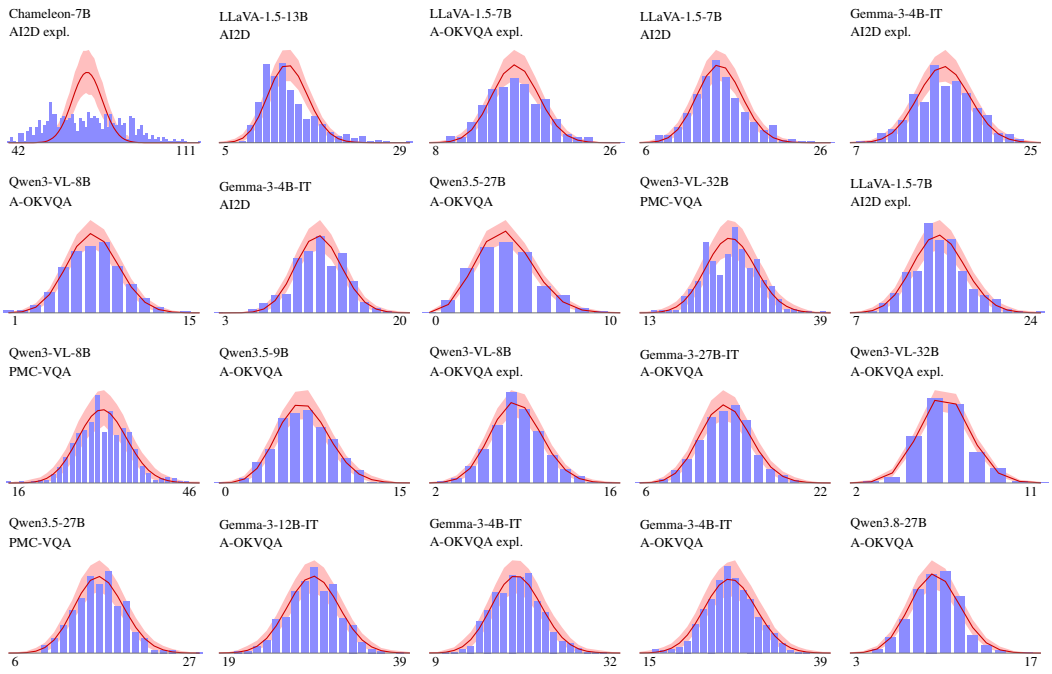

Figure 2: Rescue-count distributions for all 20 conditions with weaker letter copying. Blue bars show the number of demonstrations at each rescue count. The red curve shows the expected distribution under the exchangeable null, and the shaded band indicates the 95% range under this null. Axis endpoints indicate the minimum and maximum rescue counts in each condition.

and measure

$$G_t = 100 \left(\hat{p}_{B_t j_t^*} - \frac{1}{D} \sum_j \hat{p}_{B_t j} \right).$$

The reported rescue-rate gain is the mean of G_t over 200 random splits, in percentage points. Positive values mean that the selected demonstration beats the mean candidate on held-out errors. The 2.5th and 97.5th percentiles measure sensitivity to splitting the fixed matrix. An interval entirely above zero means that the advantage persists across these splits.

Letter-copying score and threshold. Let $L_{ij} = \mathbf{1}\{y_i = y_j^{\text{demo}}\}$ identify pairs whose query and demonstration gold letters match. Let $c = \text{mean}_{ij}\, L_{ij}$ be the cohort match rate and $s = \text{mean}_{ij:X_{ij}=1}\, L_{ij}$ be the match rate among rescues. The reported score is

$$C_{\text{letter}} = 100\, \text{clip}_{[0,1]} \left(\frac{s - c}{1 - c} \right).$$

The clipping operator truncates values below zero to zero and values above one to one. Zero means no positive excess over the cohort chance rate, including a negative excess clipped to zero. One hundred means that every rescuing pair has matching gold letters. We use 25% as a descriptive threshold between stronger and weaker letter copying. The highest score below this threshold is 21.6%, and the lowest score above it is 30.8%. Seven conditions have $C_{\text{letter}} \geq 25\%$, and twenty fall below it.

Complete heterogeneity and selection results. Tab. 13 reports all 27 answer-letter conditions. The eight baseline conditions in Tab. 4 are included here, followed by the remaining baseline and explicit conditions and the PMC-VQA conditions. Both instruction wordings use answer-letter prompting.

Observed and null rescue-count distributions. Figure 2 shows the observed and null rescue-count distributions for all 20 conditions with weaker letter copying. Nineteen have $\tau < 1$. Chameleon-7B on AI2D with explicit wording is the sole exception.

Table 13: Complete demonstration heterogeneity and held-out selection results for all 27 conditions, grouped by dataset and model, with baseline and explicit instruction wording adjacent. All conditions use answer-letter prompting. Letter copying is the chance-adjusted score defined in Append. C.2. Bold scores mark values at or above 25%. This score is distinct from the letter-associated residual variance share. Rescue-rate gain and the bracketed split intervals are in percentage points. Brackets give the 2.5th–97.5th percentiles over 200 splits. – denotes an unreported interval. Bold gains have intervals entirely above zero.

Model	Wording	Letter copying (%)	Disp.	Rescue-rate gain (pp)	Split interval	VR	KS	10-NN enrichment	τ
A-OKVQA									
Gemma-3-4B-IT	Baseline	0.0	0.96	-0.5	$[-6.3, 5.5]$	0.913	0.0191	0.82	0.00
	Explicit	0.7	0.97	$+1.2$	$[-5.5, 7.2]$	0.948	0.0186	1.20	0.00
Gemma-3-12B-IT	Baseline	0.0	0.98	-0.5	$[-6.6, 5.9]$	0.954	0.0126	0.96	0.00
Gemma-3-27B-IT	Baseline	0.0	1.01	-0.9	$[-7.9, 4.2]$	1.017	0.0246	1.46	0.14
Gemma-4-12B	Baseline	**55.0**	2.62	$+6.0$	–	6.870	0.3312	3.63	2.42
Chameleon-30B	Baseline	**49.0**	3.90	$+\mathbf{21.3}$	$[14.0, 27.7]$	15.168	0.5119	7.31	3.77
LLaVA-1.5-7B	Explicit	0.0	1.23	$+5.4$	$[-4.0, 12.5]$	1.524	0.0645	1.69	0.72
Qwen3-VL-8B	Baseline	0.0	1.13	$+1.6$	$[-5.1, 8.2]$	1.280	0.0321	1.28	0.53
	Explicit	1.7	1.02	$+0.7$	$[-6.5, 7.2]$	1.040	0.0175	1.55	0.20
Qwen3-VL-32B	Explicit	2.1	1.00	$+3.4$	$[-8.0, 10.5]$	1.005	0.0162	1.14	0.00
Qwen3.5-9B	Baseline	21.6	1.04	-1.9	$[-7.5, 4.1]$	1.072	0.0251	1.27	0.29
	Explicit	**35.3**	1.10	$+1.5$	$[-8.8, 10.5]$	1.215	0.0426	3.08	0.46
Qwen3.5-27B	Baseline	0.0	1.12	$+1.3$	$[-5.1, 8.2]$	1.257	0.0402	1.38	0.50
Qwen3.8-27B	Baseline	7.8	0.94	$+1.4$	$[-7.6, 9.8]$	0.889	0.0326	1.00	0.00
AI2D									
Gemma-3-4B-IT	Baseline	1.1	1.12	-1.0	$[-9.4, 6.0]$	1.258	0.0585	1.62	0.50
	Explicit	0.0	1.15	$+0.2$	$[-9.4, 6.6]$	1.325	0.0486	2.31	0.57
Gemma-4-12B	Baseline	**60.9**	2.15	$+7.2$	–	4.592	0.2407	3.19	1.90
Chameleon-7B	Baseline	**40.6**	3.67	$+\mathbf{16.5}$	$[7.1, 22.5]$	13.479	0.3279	1.84	3.53
	Explicit	10.0	2.53	$+\mathbf{8.9}$	$[3.0, 13.9]$	6.410	0.2615	1.35	2.32
Chameleon-30B	Baseline	**40.6**	1.20	$+2.3$	$[-4.3, 7.7]$	1.429	0.1524	5.95	0.66
LLaVA-1.5-7B	Baseline	1.9	1.23	$+2.4$	$[-2.5, 6.6]$	1.494	0.0475	2.45	0.72
	Explicit	1.6	1.06	$+1.3$	$[-2.4, 4.4]$	1.127	0.0214	2.11	0.35
LLaVA-1.5-13B	Baseline	0.1	1.37	$+6.2$	$[-2.2, 12.0]$	1.879	0.1099	4.87	0.94
PMC-VQA									
Qwen3-VL-8B	Baseline	0.6	1.06	$+0.5$	$[-4.1, 3.8]$	1.125	0.0248	1.16	0.35
Qwen3-VL-32B	Baseline	0.0	1.08	$+0.6$	$[-4.5, 4.9]$	1.152	0.0536	1.85	0.41
Qwen3.5-9B	Baseline	**30.8**	1.40	$+4.4$	–	1.942	0.0967	2.90	0.98
Qwen3.5-27B	Baseline	0.0	1.00	$+1.7$	$[-3.5, 6.1]$	1.002	0.0126	1.15	0.00

Sensitivity to cohort sampling. For Gemma-3-4B-IT on A-OKVQA under baseline answer-letter prompting, we repeat the 500×500 evaluation on two additional query–demonstration cohorts. Their query/demonstration seed pairs are $(20260902, 20260901)$ and $(20260903, 20260904)$, respectively. Records do not overlap across cohorts, although images may overlap. Tab. 14 compares the three cohorts. VR and Disp. remain below one, and $\tau = 0$ in all three: so resampling does not change the qualitative conclusion for conditions.

Table 14: Sensitivity to 500×500 cohort sampling for Gemma-3-4B-IT on A-OKVQA under baseline answer-letter prompting.

Cohort	KS	VR	Disp.	τ
Original	0.0191	0.9129	0.9580	0.00
Resample A	0.0188	0.8974	0.9491	0.00
Resample B	0.0307	0.8325	0.9129	0.00

C.3 Content similarity and answer-letter controls

Binary pairing residual. Starting from the binary rescue matrix, we iteratively remove queries that no demonstration rescues or that every demonstration rescues, and demonstrations that rescue no query or every query. These constant rows and columns carry no pairing information. On the retained matrix, the binary pairing residual is

$$R^X_{ij} = X_{ij} - \bar{X}_{i\cdot} - \bar{X}_{\cdot j} + \bar{X}_{\cdot\cdot}$$

It removes query rescue rates and demonstration rescue rates before measuring pair-specific association. This residual is binary-derived and is distinct from the continuous residual R above.

Linear content alignment. The alignment analysis in Tabs. 4 and 15 uses two content representations. BM25 is the Okapi score between the two question texts, with k_1=1.5 and b=0.75. CLIP is the cosine between the two images in the image tower of `openai/clip-vit-large-patch14`. Let S be a BM25 or CLIP query–demonstration similarity matrix on the same retained rows and columns, and let

$$\widetilde{S}_{ij} = S_{ij} - \bar{S}_{i\cdot} - \bar{S}_{\cdot j} + \bar{S}_{\cdot\cdot}$$

The reported alignment share is

$$A_S = \frac{\langle R^X, \widetilde{S} \rangle^2}{\|R^X\|_F^2 \|\widetilde{S}\|_F^2}.$$

It is the squared correlation between vectorized double-centred matrices. Zero means no linear alignment with that measured similarity. Larger values mean stronger linear alignment.

Answer-letter association. We group the retained queries and demonstrations separately by their gold answer letters. For each side, we construct a group-indicator matrix: each column represents one gold letter, and an entry is one if the query or demonstration belongs to that group and zero otherwise. We subtract each column's mean and use singular value decomposition to obtain an orthonormal basis for the remaining nonzero directions. Let U_q and U_d denote these bases for queries and demonstrations, respectively. The corresponding projection matrices are

$$P_q = U_q U_q^\top, \qquad P_d = U_d U_d^\top.$$

Because R^X is already double-centred, $P_q R^X P_d$ replaces each entry with the average residual of pairs having the same query gold letter and the same demonstration gold letter. For example, all pairs with query gold letter A and demonstration gold letter B receive the mean residual of that group. This keeps the differences associated with gold-letter combinations and removes differences within each group.

The letter-associated residual variance share is

$$A_L = \frac{\|P_q R^X P_d\|_F^2}{\|R^X\|_F^2},$$

where the squared Frobenius norm is the sum of the squared matrix entries. Thus, A_L measures the fraction of residual variation captured by these gold-letter groups. Zero means that the group means contain no residual variation, and larger values indicate stronger answer-letter association. This measures association with gold-letter combinations, not how often the model copies a demonstration's answer letter. With all four letters present on both sides, each centred grouping has three independent directions, giving nine interaction directions in total. Each content-similarity matrix provides only one direction, so their variance shares should be interpreted using the separate chance references below.

Chance references. The three projections share the same binary rescue residual denominator, but differ in their effective degrees of freedom. Each double-centred similarity matrix provides one direction. With four gold-letter groups on each side, the letter interaction spans $(4-1)(4-1) = 9$ directions. Across these conditions, the chance reference for the fraction of residual variance explained is approximately 0.0007%–0.0095% for both BM25 and CLIP, and 0.0066%–0.0859% for answer-letter association. These are the reference shares that unrelated predictors can capture by chance.

Table 15: Percentage of binary rescue residual variance explained by answer-letter groups (Letter), BM25, and CLIP across all 27 conditions. Dataset panels and model rows follow Tab. 13. Wording distinguishes baseline and explicit instructions, both under answer-letter prompting. The printed precision is retained from the original results, so 0.000 denotes a rounded value. Chance references are given above.

Model	Wording	Explained residual variance (%)		
		Letter	BM25	CLIP
A-OKVQA				
Gemma-3-4B-IT	Baseline	1.14	0.000	0.059
	Explicit	0.18	0.027	0.059
Gemma-3-12B-IT	Baseline	0.26	0.000	0.001
Gemma-3-27B-IT	Baseline	0.26	0.012	0.083
Gemma-4-12B	Baseline	49.77	0.000	0.000
Chameleon-30B	Baseline	24.71	0.001	0.000
LLaVA-1.5-7B	Explicit	0.13	0.009	0.034
Qwen3-VL-8B	Baseline	0.03	0.015	0.063
	Explicit	0.18	0.000	0.019
Qwen3-VL-32B	Explicit	1.03	0.001	0.000
Qwen3.5-9B	Baseline	4.78	0.000	0.010
	Explicit	16.38	0.026	0.013
Qwen3.5-27B	Baseline	0.32	0.041	0.003
Qwen3.8-27B	Baseline	1.68	0.001	0.022
AI2D				
Gemma-3-4B-IT	Baseline	0.43	0.071	0.104
	Explicit	0.72	0.060	0.442
Gemma-4-12B	Baseline	49.57	0.000	0.003
Chameleon-7B	Baseline	17.31	0.004	0.010
	Explicit	2.19	0.000	0.021
Chameleon-30B	Baseline	30.08	0.000	0.020
LLaVA-1.5-7B	Baseline	0.12	0.437	0.314
	Explicit	0.22	0.359	0.398
LLaVA-1.5-13B	Baseline	0.06	0.266	0.299
PMC-VQA				
Qwen3-VL-8B	Baseline	0.05	0.000	0.008
Qwen3-VL-32B	Baseline	0.02	0.009	0.060
Qwen3.5-9B	Baseline	8.42	0.009	0.047
Qwen3.5-27B	Baseline	2.21	0.019	0.135

Cohesion among rescuing demonstrations. For an eligible query i, let H_i and N_i be its rescuing and non-rescuing candidate sets. Let G_{uv} be the cosine similarity between candidates u and v in one representation space. We compute

$$B_i = \frac{1}{|H_i|(|H_i|-1)} \sum_{\substack{u,v \in H_i \\ u \neq v}} G_{uv}, \qquad O_i = \frac{1}{|H_i||N_i|} \sum_{u \in H_i} \sum_{v \in N_i} G_{uv}.$$

The analysis uses queries with at least two rescues and one non-rescue. The condition-level Both rescue and One rescues means average B_i and O_i over eligible queries. Tab. 16 reports their medians across conditions. Largest gap is the maximum condition-level difference $B - O$. A positive gap means that rescuing demonstrations are more similar to one another on average. It does not show that similarity can rank individual candidates accurately.

The representations. The content-structure analysis in Tab. 16 uses CLIP images and four additional representations. SigLIP (Zhai et al., 2023) is `google/siglip-base-patch16-224` and DINOv2 (Oquab et al., 2024) is `facebook/dinov2-base`, both read from the image tower, and DINOv2 from the class token because it ships no projection head. BGE (Xiao et al., 2024) is `BAAI/bge-base-en-v1.5` with class-token pooling and no instruction prefix. BridgeTower (Xu et al., 2023) encodes the image and the text of a candidate together. The representation choices are fixed before the content-structure analysis.

Content structure among rescuing demonstrations. Tab. 16 covers the 27 answer-letter conditions and three option-text controls. In all five spaces, the median similarity between two rescuing demonstrations is slightly higher than that between a rescuing and a non-rescuing demonstration. The median gaps are at most 0.026, and the largest within-condition gap is 0.083. The representations therefore capture slight content cohesion but provide little separation for accurate retrieval or ranking.

Table 16: Similarity among demonstrations that rescue the same query across five representation spaces, 27 answer-letter conditions, and three option-text controls. Both rescue compares two rescuing demonstrations. One rescues compares a rescuing and a non-rescuing demonstration. Both similarity columns are medians across conditions. Largest gap is the maximum across conditions of Both rescue minus One rescues.

Representation	Both rescue	One rescues	Largest gap
BridgeTower	0.878	0.872	0.014
SigLIP image	0.580	0.561	0.057
CLIP image	0.553	0.534	0.068
BGE text	0.487	0.474	0.083
DINOv2 image	0.182	0.157	0.083

Option-text control. Under option-text prompting, original letters remain in cohort metadata but are never shown to the model. The chance-adjusted letter-copying score is defined in Append. C.2. Tab. 5 reports demonstration heterogeneity, held-out rescue-rate gains, and letter-copying scores under both answer formats. Tab. 17 additionally reports zero-shot accuracy and rescue rate. Rescue rate is the fraction of zero-shot errors corrected by a uniformly drawn candidate demonstration, averaged over queries.

Table 17: Letter-copying scores, zero-shot accuracy, and rescue rates under answer-letter and option-text prompting. All values are percentages. Paired values correspond to answer-letter and option-text prompting, respectively.

Condition	Letter-copying score	Zero-shot accuracy	Rescue rate
Gemma-4-12B / A-OKVQA	55.0 → 0.8	65.0 → 69.0	30.3 → 16.0
Gemma-4-12B / AI2D	60.9 → 0.0	82.0 → 84.6	28.9 → 18.2
Qwen3.5-9B / PMC-VQA	30.8 → 0.4	61.8 → 60.6	12.5 → 11.0

C.4 QUERY-LEVEL CASES

Setup and queries. To complement the aggregate results in Sec. 4.1, we run a fixed-pool one-shot sweep with LLaVA-1.5-7B. Tab. 18 reports eight validation-set zero-shot errors, four each from A-OKVQA and ScienceQA (Lu et al., 2022). BridgeTower first ranks training-set demonstrations for each validation query. Using seed 42, we sample two errors per dataset that its top ten cannot rescue and two that at least one rescues. The sweep also includes one zero-shot success per dataset to test for broken answers. These rows are omitted, but their retrieved candidates enter the shared pools below.

Demonstration pools. Each dataset shares 5000 labeled training-split demonstrations. Each comprises a question, an image when present, answer options, and the gold answer. The pool starts with the union of the top 50 candidates returned for the five selected queries by BridgeTower, CLIP image, SigLIP image, SigLIP text–image, DINOv2 image, and BGE text. Random demonstrations sampled from the remaining training split with seed 43 fill the pool to 5000. Each error is evaluated once with every pool demonstration. The Retrieved subset is the top-50 union, and the Random subset is the randomly sampled remainder.

Results. The full sweep separates four queries rescued by at most two of the 5000 demonstrations from four whose rescue rates range from 0.08% to 75.14%. We label these groups insensitive and sensitive, respectively. Within the sensitive group, the retrieved and random subsets have similar rescue rates. Together, these cases show that rescue depends strongly on the query itself.

Table 18: Eight validation-set zero-shot errors of LLaVA-1.5-7B, each paired separately with every demonstration in its dataset's fixed pool of 5000 training-set demonstrations. Rescued counts wrong-to-right changes. Retrieved subset uses the dataset-level six-selector top-50 union shared across queries. Random subset uses the randomly filled remainder. A query is insensitive when almost no demonstration rescues it and sensitive when some do and some do not.

#	Dataset	Regime	Rescued / 5000	Retrieved subset	Random subset
1	A-OKVQA	insensitive	0 (0.00%)	0.00%	0.00%
2	A-OKVQA	insensitive	2 (0.04%)	0.00%	0.05%
3	ScienceQA	insensitive	0 (0.00%)	0.00%	0.00%
4	ScienceQA	insensitive	0 (0.00%)	0.00%	0.00%
5	ScienceQA	sensitive	4 (0.08%)	0.61%	0.00%
6	ScienceQA	sensitive	339 (6.78%)	5.50%	6.97%
7	A-OKVQA	sensitive	1762 (35.24%)	36.72%	34.88%
8	A-OKVQA	sensitive	3757 (75.14%)	72.10%	75.88%

D END-TO-END DEMONSTRATION USE

D.1 EVALUATION PROTOCOL

Conditions and data splits. The end-to-end comparison evaluates Gemma-3-4B-IT on A-OKVQA and LLaVA-1.5-13B on AI2D. Both conditions use the baseline answer-letter prompts defined above. A-OKVQA provides 17,056 training queries and 1,145 validation queries. AI2D provides 7,784 training queries and 1,897 official test queries. Tab. 19 lists the disjoint roles used to fit and tune the gate and arbitration procedure. Every row of Tabs. 6 and 7 draws demonstrations from the full training split of its condition and scores the official evaluation split once.

Table 19: Query roles in the gate and arbitration evaluation. Selection, fit, and dev are disjoint portions of the training split. The selection set is the first 2,000 queries in loader order. Fit and dev use a fixed permutation with seed 20260908.

Dataset	Role	n	Construction	Use
A-OKVQA	Selection	2,000	First 2,000 training queries	Choose m=16 and r=25%
	Fit	12,045	80% of the remaining 15,056 after permutation	Fit the gate
	Dev	3,011	Remaining 20%	Early stopping and gate-score cutoff
	Evaluation	1,145	Official validation split	Final report
AI2D	Selection	2,000	First 2,000 training queries	Choose m=2 and r=10%
	Fit	4,627	80% of the remaining 5,784 after permutation	Fit the gate
	Dev	1,157	Remaining 20%	Early stopping and gate-score cutoff
	Evaluation	1,897	Official test split	Final report

Number of demonstrations and prompt templates. The number k denotes how many demonstrations are included in a single prompt. With $k = 1$, one demonstration precedes the final query; with $k = 4$, four demonstrations precede it in the same prompt, rather than being evaluated in four separate prompts. Each demonstration contains its image, question, answer options, and gold answer letter. The final query contains the same fields except for the answer.

The complete four-demonstration prompt body is shown below. The model's `USER:` and `ASSISTANT:` delimiters surround this body. The opening instruction wraps here for readability but occupies one line in the actual prompt.

```
Use the solved examples to answer the final multiple-choice
visual question.
Each image token corresponds to the image immediately below it.

Example 1:
<image>
Question: <demonstration 1 question>
Choices:
A. <demonstration 1 option A>
B. <demonstration 1 option B>
C. <demonstration 1 option C>
D. <demonstration 1 option D>
Answer: <demonstration 1 gold letter>

Example 2:
<image>
Question: <demonstration 2 question>
Choices:
A. <demonstration 2 option A>
B. <demonstration 2 option B>
C. <demonstration 2 option C>
```

```
D. <demonstration 2 option D>
Answer: <demonstration 2 gold letter>

Example 3:
<image>
Question: <demonstration 3 question>
Choices:
A. <demonstration 3 option A>
B. <demonstration 3 option B>
C. <demonstration 3 option C>
D. <demonstration 3 option D>
Answer: <demonstration 3 gold letter>

Example 4:
<image>
Question: <demonstration 4 question>
Choices:
A. <demonstration 4 option A>
B. <demonstration 4 option B>
C. <demonstration 4 option C>
D. <demonstration 4 option D>
Answer: <demonstration 4 gold letter>

Final question:
<image>
Question: <query question>
Choices:
A. <query option A>
B. <query option B>
C. <query option C>
D. <query option D>
Answer with the option letter only.
```

For $k = 1$, the prompt retains only Example 1 and the final question; the opening and answer instructions are unchanged. Gemma-3-4B-IT uses the same demonstration and query fields, passed as one user turn through its native chat template.

Retrieval baselines. The retrievers follow Sec. 5. MMICES keeps the top $K{=}50$ candidates by image similarity before text reranking. Each training-free retriever runs at $k{=}1$ and $k{=}4$. Random single reports mean accuracy over the same 20 separate draws. MSIER is trained as described below.

Training the retriever. A frozen CLIP scores every candidate by the cosine between the query image and the candidate image plus the cosine between the query image and the candidate text. We retain the 50 highest-scoring candidates. The answering model of that condition scores each of the 50 once by forced-choice letter log-probability. The best five become positives and the worst five become negatives, and we fine-tune the CLIP bi-encoder on them with an in-batch contrastive loss. MSIER partitions all training queries with the $80/20$ rule and seed 20260908. This gives $13{,}645$ fit and $3{,}411$ dev queries on A-OKVQA, and $6{,}227$ fit and $1{,}557$ dev queries on AI2D.

The gate's inputs. The gate reads seven confidence features derived from the four letter log-probabilities. They are the four values sorted together with the top-two margin, the entropy and the top probability. The gate concatenates those seven with the answer-position hidden states from four layers, counted back from the last as -1, -5, -9 and -13 on both conditions. Its input is therefore $10{,}247$ wide on Gemma-3-4B-IT and $20{,}487$ wide on LLaVA-1.5-13B.

Training the gate. The gate is a logistic regression with an L_2 penalty. Its inputs are the features of the paragraph above, and its label is that the query will be answered incorrectly at zero shot. The fit and dev sets are listed in Tab. 19. We evaluate zero-shot error detection by the area under the receiver operating characteristic curve (AUROC), treating errors as the positive class. A value of 0.5 indicates random ranking and 1 perfect discrimination. On A-OKVQA, the gate achieves AUROCs of 0.8239 on the development set, 0.8099 on the parameter-selection set, and 0.8215 on the evaluation set. The corresponding values on AI2D are 0.8066, 0.7895 and 0.8122.

Choosing m and r. We select m and r by evaluating a grid of 209 combinations. m runs from 2 to 20 in steps of one. r runs from 5% to 40% in steps of five points and then takes 50%, 75% and 100%, where 100% sends every query to the arbitration. Both conditions run the whole grid on their selection set in Tab. 19, which plays no role in fitting the gate, and keep the combination with the largest accuracy gain over zero shot there. It returns m=16 and r=25% on A-OKVQA, and m=2 and r=10% on AI2D. The evaluation split of a condition is then scored once with those two values. Append. D.2 defines the development-derived cutoff.

Inference cost. A retrieval prediction uses one forward pass of the answering model per query. Gate and arbitration uses one zero-shot forward pass on every query and 20 additional one-shot forward passes on each gated query. The gate fires on 19.7% of the A-OKVQA evaluation queries and 10.5% of the AI2D evaluation queries, giving averages of approximately 4.9 and 3.1 answering-model forward passes per query, respectively.

Removing the gate. Running arbitration on every query instead of only the gated queries gives $+0.44$ pp on A-OKVQA and -1.48 pp on AI2D, compared with $+1.22$ and $+0.53$ pp for gate and arbitration. All changes are relative to zero-shot accuracy.

D.2 Gate, arbitration, and outcome definitions

Gate and requested trigger rate. The gate assigns score s_i for the event $z_i \neq y_i$. For a requested trigger fraction r, the development scores set the absolute cutoff

$$q_r = \text{quantile}_{1-r}\{s_i : i \in \text{dev}\}.$$

The gate fires on a query when $s_i \geq q_r$. The same cutoff is then applied to the selection and evaluation sets.

Arbitration. For each fired query, let a_{it} be the prediction under random demonstration $t \in \{1, \ldots, 20\}$. The zero-shot prediction z_i is the incumbent. For each alternative answer $a \neq z_i$, define its challenger votes as

$$v_i(a) = \sum_{t=1}^{20} \mathbf{1}\{a_{it} = a\}.$$

Let a_i^* be the unique alternative with the most votes. The final answer is

$$\widehat{z}_i = \begin{cases} a_i^*, & \text{if the gate fires, } v_i(a_i^*) \geq m, \text{ and the top is unique,} \\ z_i, & \text{otherwise.} \end{cases}$$

Incumbent predictions are excluded from the challenger count. A tie among top challengers keeps the incumbent.

Rescues, breaks, and net accuracy. For the final predictions, define

$$\text{Rescued} = \sum_i \mathbf{1}\{z_i \neq y_i,\ \widehat{z}_i = y_i\}, \qquad \text{Broken} = \sum_i \mathbf{1}\{z_i = y_i,\ \widehat{z}_i \neq y_i\}.$$

On N evaluation queries, the accuracy change from zero shot is

$$\Delta\text{Acc.} = 100\frac{\text{Rescued} - \text{Broken}}{N}$$

in percentage points. Rescues are corrected zero-shot errors. Breaks are initially correct answers made incorrect. Equal net gains can therefore arise from different amounts of answer revision.

D.3 Sensitivity to resampling

We repeat the Gemma-3-4B-IT experiment on A-OKVQA reported in Tab. 6 with two additional seeds. We redraw disjoint selection, gate-fit, and development sets without replacement, keeping their sizes unchanged, then refit the gate and reselect m and r using the original procedure.

The sampling seed remains fixed at 20260908, but the demonstrations sampled for selection queries change because the selection set changes. All runs use the same 1,145 validation queries and

the same 20 randomly sampled demonstrations per validation query, all other settings remain unchanged.

Tab. 20 compares the original configuration with the two resampled runs.

Table 20: Gate and arbitration under resampling for Gemma-3-4B-IT on A-OKVQA. Numeric row labels give the seed S for selection-set sampling and fit/dev splitting. Requested r is the target trigger fraction; actual is the fraction of evaluation queries on which the gate fires. Both are percentages. Net is Rescued minus Broken, and Δ is the accuracy change from zero shot in percentage points.

Configuration	m	Requested r (%)	Actual (%)	Rescued	Broken	Net	Δ (pp)
Original	16	25	19.7	18	4	+14	+1.22
68129996	15	20	15.1	18	5	+13	+1.14
21268241	12	30	25.8	31	20	+11	+0.96