FROM PAIRWISE EVIDENCE TO SET UTILITY: A CONTROLLED META-EVALUATION OF HOMOGENEITY MEASURES FOR LLM RESPONSE SELECTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Selecting a small subset of large language model responses requires removing redundancy while preserving distinct valid alternatives. However, strong pairwise homogeneity prediction for the intended use does not guarantee good subset selection; even with perfect pairwise rankings, the selection pipeline can still incur avoidable coverage loss under a fixed selection budget. To explain this gap, we introduce a controlled meta-evaluation framework that holds responses fixed while varying their intended use across four task families with verifiable answers. We trace how decision thresholds fixed on development data interact with clustering and budgeted selection to shape final coverage. When evaluation pools different uses, a purpose-only control can achieve high AUC without inspecting response content, yet cannot distinguish alternatives within the target use. Exhaustively enumerating input orders further shows how repetition patterns, ordering, and the selection budget affect whether incorrect merges or splits reduce coverage. On Natural-50, a collection of open-ended responses with human-annotated functional units, we evaluate coverage when retaining three of six responses per query. Transferred to 30 test queries, the pipeline selected by pooled AUC on controlled tasks incurs a mean per-query coverage shortfall of 8.36 percentage points relative to the optimum attainable from the same candidates under the same budget (95% CI: 1.94–16.03), with losses occurring in six queries. These findings call for evaluating homogeneity metrics for response selection by jointly examining pairwise performance for the intended use, functional coverage, and avoidable loss under the actual selection budget.

1 INTRODUCTION

Users often request multiple responses from language models to obtain different ideas, solutions, or options. Yet more responses do not necessarily yield more useful content. Jiang et al. Jiang et al. (2025) observed substantial convergence within and across models on 26,000 open-ended questions, a phenomenon they termed the *Artificial Hivemind*. When only a few responses can be shown, the practical question is therefore not simply how diverse generation is, but whether selection removes redundancy without discarding distinct useful alternatives.

Consider a route-planning assistant that produces four responses: two describe the same subway route and two recommend the same taxi option. If only two responses can be retained, choosing both subway responses preserves one practical choice, whereas choosing one subway and one taxi response preserves two. Difference alone is not sufficient, however: a route that violates the arrival-time constraint should not count as a useful alternative. Selection must therefore preserve distinctions that are both valid and relevant to the intended use.

Pairwise homogeneity measures offer one way to identify such distinctions, but subset selection requires more than ranking response pairs. Scores must be thresholded into relation decisions, aggregated into groups, and converted into representatives under a fixed budget. A false equivalence can merge distinct alternatives; an incorrect split can allow redundant responses to occupy multiple slots; and either error may or may not reduce final coverage depending on linkage, response order,

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.

repetition, and budget. Strong pairwise ranking is therefore evidence about one stage of the pipeline rather than a guarantee about the utility of the selected set.

We study this gap through three connected questions. **Intended-Use Alignment** asks whether scores reflect distinctions that matter for the target use rather than benchmark composition. **Fixed-Threshold Transfer** asks whether a threshold selected on development data continues to separate equivalent and distinct pairs on new response sets. **Selection Compatibility** asks whether those local decisions, after grouping and representative selection, preserve distinct valid content under the available budget. AUC captures relative score ordering, but it does not determine the transferred operating point or the set produced downstream.

To isolate these stages, we construct a controlled meta-evaluation in which candidate responses remain fixed while their intended use changes across four task families with programmatically verifiable answers. We propagate development-fixed thresholds through single- or complete-linkage grouping and budgeted representative selection, enumerate response orders, and vary repetition patterns and budgets in a three-alternative extension. We then evaluate the same distinction on **Natural-50**, which contains complete open-ended response sets with human annotations of validity and functional units. Functional Coverage measures the union of valid units retained by the selected set, and Avoidable Coverage Loss compares that coverage with the maximum attainable from the same candidates under the same budget.

The experiments expose two separations. First, pooled evaluation can reward use-related cues that do not discriminate alternatives within the target use: a Purpose-Only Control reaches Pooled AUC 0.875 while remaining at chance for Alternative-Preservation AUC. Second, perfect intended-use ranking can still lose attainable coverage after a development-fixed threshold and grouping are applied. On Natural-50, pairwise AUC and Functional Coverage also need not induce the same ordering of scoring interfaces, and the controlled-task Pooled-AUC transfer scheme incurs measurable avoidable loss on the final test set. Our contribution is therefore an evaluation framework—not a new homogeneity scorer—that connects use-conditioned pairwise evidence to threshold transfer, grouping, representative selection, and same-budget set utility.

2 Related Work

Diversity Evaluation. Diversity in natural language generation spans lexical, syntactic, semantic, and functional variation. Controlled studies show that decoding can alter surface form more readily than meaning (Tevet & Berant, 2021), while linguistic and distributional measures capture complementary variation (Guo et al., 2025; Pillutla et al., 2021). NLG evaluation work likewise emphasizes criterion-specific validity and measurement uncertainty (Sai et al., 2021; Xiao et al., 2023). Studies of the Artificial Hivemind and generative monoculture document convergence in model outputs (Jiang et al., 2025; Wu et al., 2025). We instead ask which distinct valid content survives downstream selection from fixed candidates.

Functional Equivalence and Use. Semantic similarity is related to, but not identical with, functional equivalence. Sentence-BERT and SimCSE measure general semantic proximity (Reimers & Gurevych, 2019; Gao et al., 2021), whereas NoveltyBench and task-anchored functional diversity ground distinctions in task-relevant meaning (Zhang et al., 2025; Jain et al., 2025). Conditional semantic similarity further makes the comparison condition explicit (Deshpande et al., 2023; Tu et al., 2024). We vary intended use while holding responses fixed and, for open-ended responses, measure the union of retained valid functional units rather than forcing partial overlaps into one equivalence class.

From Evaluators to Set Selection. Pairwise scores affect selection only after they are converted into decisions and combined across a set. Model-based judgments can vary with evaluator and presentation effects (Liu et al., 2023; Zheng et al., 2023; Wang et al., 2024). Set-level diversity and subset methods include the Vendi Score, determinantal point processes, maximal marginal relevance, and submodular objectives (Friedman & Dieng, 2023; Kulesza & Taskar, 2012; Carbonell & Goldstein, 1998; Lin & Bilmes, 2011). We focus on a different transition: inferred functional equivalence is thresholded, grouped, and converted into representatives, then evaluated against the same-budget maximum Functional Coverage.

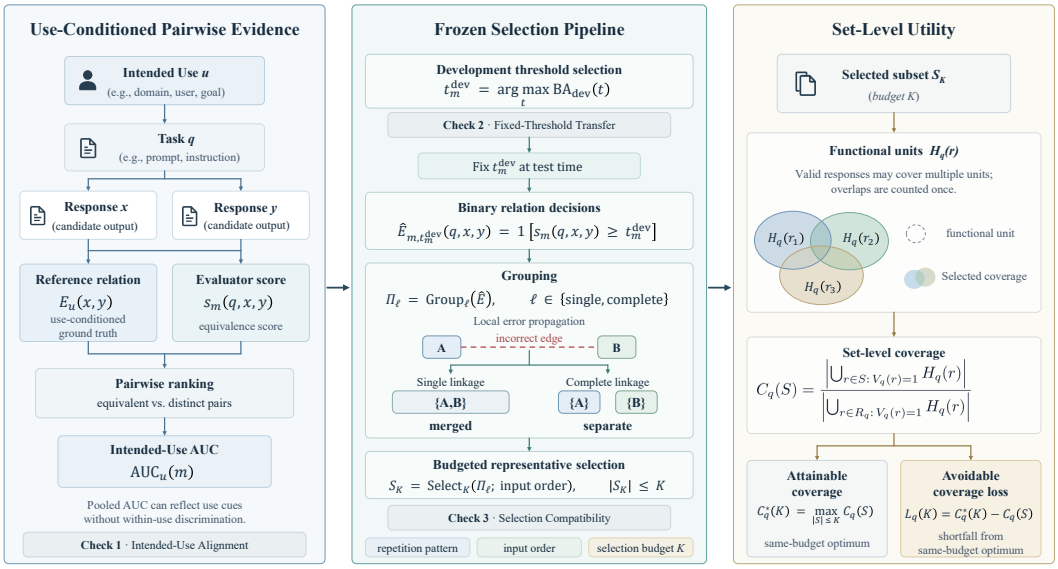


Figure 1: Meta-evaluation framework linking use-conditioned pairwise scores to set-level utility. Scores are converted into equivalence decisions using development-set thresholds, followed by grouping and budgeted representative selection. Intended-Use Alignment, Fixed-Threshold Transfer, and Selection Compatibility assess successive stages of this process. Functional Coverage and Avoidable Coverage Loss measure the distinct valid content retained by the selected set.

3 From Pairwise Evidence to Set Utility

3.1 Validity and Equivalence Depend on Intended Use

Functional equivalence depends on use. Both "73" and "89" satisfy a request for a prime number between 70 and 100: either confirms feasibility, but they constitute distinct alternatives when the aim is to preserve options. "73" and "73.0" represent the same option under both uses. For controlled case c, let $V_c(x)$ denote response validity and $a_c(x)$ its canonical object, independent of wording. For valid x, y, define

$$E_c^{\mathrm{wit}}(x,y) = 1, \qquad E_c^{\mathrm{alt}}(x,y) = \mathbf{1}[a_c(x) = a_c(y)]\,. \tag{1}$$

Here wit denotes **Feasibility-Confirmation Use**, and alt denotes **Alternative-Preservation Use**. Relations involving invalid responses are handled separately; the controlled retention experiments evaluate alternative preservation.

3.2 Ranking and Threshold Performance Must Be Interpreted with Respect to Use

Evaluator m assigns a score $s_m(q,x,y) \in [0,1]$ to a valid response pair, with higher scores indicating equivalence. Equivalent and distinct pairs form the positive and negative classes. AUC measures score ranking; binary decisions use

$$\widehat{E}_{m,t}(q,x,y) = \mathbf{1}[s_m(q,x,y) \geq t]\,. \tag{2}$$

The threshold t maximizes development-set Balanced Accuracy (BA), the mean of Equivalent Acceptance Rate and Distinct Rejection Rate, and remains fixed on test sets. Perfect ranking can still yield incorrect binary decisions at a transferred threshold. For pooled uses, let p_u^+ and p_v^- be the class-normalized weight proportions from uses u and v, and $\widehat{\mathrm{AUC}}_{uv}$ the conditional AUC comparing their positive and negative pairs:

$$\widehat{\mathrm{AUC}} = \sum_{\substack{u,v \\ p_u^+>0,\ p_v^->0}} p_u^+ p_v^- \, \widehat{\mathrm{AUC}}_{uv}. \tag{3}$$

Cross-use comparisons ($u \neq v$) can reward use-related cues without testing discrimination among alternatives within the target use. We therefore report Alternative-Preservation Intended-Use AUC separately. All pairwise measures use equal case weighting within each class; estimators and threshold-selection details appear in Appendices A.1 and C.3.

3.3 Grouping and Selection Determine Final Functional Coverage

Thresholded equivalence scores define pairwise edges. Single linkage groups responses by connectivity; complete linkage merges clusters only when every cross-cluster pair meets the threshold. For budget K, clusters are ordered by their earliest input position and represented by that response. We retain the first K representatives, filling unused slots with unselected responses in input order. Incorrect merges or splits can thus redirect selection slots.

In controlled retention, all candidates are valid and each response maps to one canonical object. For valid candidates X_c^{ret} and selected responses B_c, define

$$U_c = \frac{|\{a_c(x) : x \in B_c\}|}{|\{a_c(x) : x \in X_c^{\mathrm{ret}}\}|}, \qquad L_c = U_c^* - U_c. \tag{4}$$

Here U_c^* is the same-budget attainable maximum: for M distinct objects, $U_c^* = \min(K, M)/M$. The resulting loss L_c excludes the coverage limit imposed by K (Appendix A.3).

For open-ended responses, let R_q be query q's candidates, $V_q(r)$ their validity, and $H_q(r)$ the functional units in response r. For the selected subset $S \subseteq R_q$, define

$$C_q(S) = \frac{\left|\bigcup_{r \in S,\ V_q(r)=1} H_q(r)\right|}{\left|\bigcup_{r:\ V_q(r)=1} H_q(r)\right|}, \qquad C_q^*(K) = \max_{|S| \leq K} C_q(S). \tag{5}$$

Overlapping valid units are counted once; invalid responses occupy slots but contribute no units. We enumerate feasible subsets to obtain $C_q^*(K)$ and report avoidable loss as $C_q^*(K) - C_q(S)$.

3.4 Experimental Setup

The controlled suite contains four task families (prime numbers, subset sum, shortest path, scheduling), 64 development cases, and 128 test cases. Original cases contain four valid responses representing two alternatives ($K = 2$); a three-alternative extension varies repetition and budget. Natural-50 provides six annotated responses per query: after one prespecified exclusion, 10 queries are used for threshold development, 9 for scheme selection, and 30 for final testing ($K = 3$).

We evaluate Jaccard, TF-IDF, STS, NoveltyBench, and two interfaces sharing quantized Qwen3-4B-Instruct-2507 weights: Qwen3-P receives task and use information, whereas Qwen3-U receives neither. Thresholds and schemes are fixed from development data. Controlled coverage averages input orders within each independent task; Natural-50 uses the frozen blinded order and independent queries. Purpose-Only Control and Canonicalization + Purpose Rule are post-hoc interpretive controls excluded from scheme selection (Appendices B–C).

4 Results

4.1 Pooled AUC Can Obscure Intended-Use Discrimination

The **Purpose-Only Control** scores all Feasibility-Confirmation pairs as 1 and all Alternative-Preservation pairs as $1/3$, the latter use's development-set equivalent-pair proportion. It ignores

Table 1: Six scoring interfaces and two interpretive controls. Pooled AUC combines both uses; Intended-Use AUC and Functional Coverage are evaluated under the Alternative-Preservation Use. Coverage uses complete linkage, development-fixed thresholds, and $K = 2$. All 24 input orders are averaged within each case.

Scoring interface / control	**Pooled AUC**	**Intended-Use AUC**	**Functional Coverage (%)**
Jaccard	0.67340	0.84679	93.00
TF-IDF	0.67932	0.85864	90.30
STS	0.71975	0.93950	97.01
NoveltyBench	0.72252	0.94505	97.85
Qwen3-U	0.75000	1.00000	100.00
Qwen3-P	0.99709	0.99998	100.00
Purpose-Only Control	0.87500	0.50000	83.33
Canonicalization + Purpose Rule	1.00000	1.00000	100.00

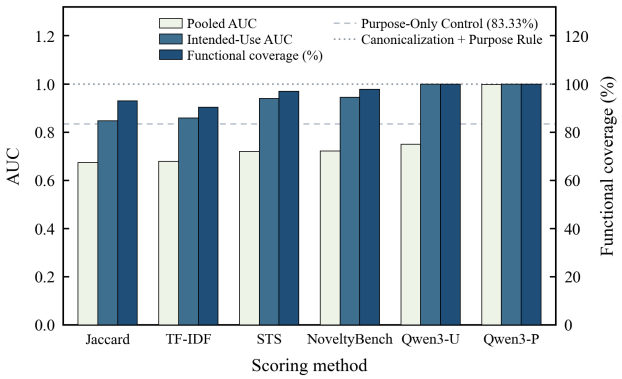


Figure 2: Pooled AUC, Intended-Use AUC, and Functional Coverage for six scoring interfaces. Horizontal lines mark the coverage of the Purpose-Only Control (83.33%) and Canonicalization + Purpose Rule (100%). The left axis shows AUC and the right axis shows coverage.

response content and assigns identical scores to equivalent and distinct pairs within the target use. Introduced post hoc for interpretation, it was excluded from scheme selection.

The control's Pooled AUC is 0.875 versus Qwen3-U's 0.750, but their Intended-Use AUCs are 0.500 versus 1.000. Under development-fixed thresholds, complete linkage, and $K = 2$, coverage is 83.33% versus 100% (Table 1; Figure 2).

By Equation (3), 75% of positive-class weight enters cross-use comparisons, perfectly ordered by the control; within-use comparisons are tied. Thus, $0.75 \times 1 + 0.25 \times 0.5 = 0.875$: high Pooled AUC here reflects use cues, not target-use alternative discrimination.

4.2 Perfect Pairwise Ranking Still Does Not Guarantee Alternative Coverage

The Pooled-AUC and Intended-Use AUC selection rules choose the highest-development-AUC interface with single linkage prespecified, yielding Qwen3-P/single and Qwen3-U/single. The Diagnostic rule selects Qwen3-P/complete by maximizing the worst-family minimum of the Equivalent Acceptance Rate and Distinct Rejection Rate; ties are resolved by development coverage, post-grouping BA, and deterministic name ordering. Thresholds and schemes are frozen for testing (Appendix C.3).

Across 128 four-response test tasks ($K = 2$), the Pooled-AUC Scheme covers 99.35% versus 100% for the Intended-Use AUC and Diagnostic Schemes (Table 2). The paired difference is 0.65 percentage points (95% CI: 0.13–1.18), with losses in 5/128 tasks. The First-K Baseline achieves 83.33%; fixed Qwen3 complete-linkage schemes attain 100%. Each task, rather than each of its 24 input orders, is an independent statistical unit.

Table 2: Order-averaged coverage (%) for development-selected schemes and simple rules. The evaluation contains 128 tasks with a selection budget of two, and the attainable coverage under the same budget is 100% for every task. Confidence intervals are obtained by resampling tasks within task families.

Scheme (model / linkage)	**Mean [95% CI]**	**Gap to optimum**	**Worst-order mean**
Pooled AUC (Qwen3-P / single)	99.35 [98.82, 99.87]	.65	98.05
Intended-Use AUC (Qwen3-U / single)	100.00 [100, 100]	.00	100.00
Diagnostic (Qwen3-P / complete)	100.00 [100, 100]	.00	100.00
Intended-Use Post-Grouping BA (Qwen3-P / complete)	100.00 [100, 100]	.00	100.00
Fixed Qwen3-P / complete	100.00 [100, 100]	.00	100.00
Fixed Qwen3-U / complete	100.00 [100, 100]	.00	100.00
First-K Baseline	83.33 [83.33, 83.33]	16.67	50.00

Table 3: Order-averaged coverage (%) for three-alternative sets. Each column uses the same 32 tasks, with 720 input orders per task. The attainable coverage is 66.67% for a selection budget of two and 100% for a selection budget of three.

Carried-over scheme	**Balanced** $K = 2$	**Balanced** $K = 3$	**Imbalanced** $K = 2$	**Imbalanced** $K = 3$
First-K Baseline	60.00	80.00	53.33	66.67
Purpose-Only Control / complete	60.00	80.00	53.33	66.67
Pooled AUC / single	66.46	97.43	66.67	98.13
Intended-Use AUC / single	66.67	99.03	66.67	98.13
Diagnostic / Intended-Use Post-Grouping BA / fixed complete	66.67	100.00	66.67	100.00
Canonicalization + Purpose Rule / fixed Qwen3-U complete	66.67	100.00	66.67	100.00

A frozen-scheme extension uses 32 six-response tasks with three alternatives, balanced repetition $(2, 2, 2)$ or imbalanced repetition $(4, 1, 1)$, and $K \in \{2, 3\}$. We average all 720 orders within each task before aggregating.

Despite Intended-Use AUC = 1, the Intended-Use AUC Scheme covers 99.03% of balanced and 98.13% of imbalanced sets at $K = 3$ (Table 3). The Pooled-AUC Scheme covers 97.43% in balanced $K = 3$ sets, a 2.57-point loss (95% CI: 0.63–4.65); the evaluated complete-linkage schemes attain the same-budget optimum. Perfect ordering does not force distinct-pair scores below a transferred threshold: a false edge can join alternatives under single linkage and cause omission during selection.

4.3 Whether Local Errors Cause Omissions Depends on Set Structure and Selection

Table 4 traces three local errors through selection. In scheduling, a false cross-alternative edge merges four responses under single linkage. With order A_1, A_2, B_1, B_2 and $K = 2$, both selected responses represent A; changing only to complete linkage preserves A and B.

The same scheduling merge preserves both alternatives under order A_1, B_1, A_2, B_2: 8/24 orders cause loss. The subset-sum case shows merges across three alternatives; the prime-number case shows an incorrect split allowing duplicate A responses to fill both slots, which changing linkage alone cannot resolve.

Repetition and budget also matter. At $K = 3$, First-K covers 80.00% with repetition $(2, 2, 2)$ versus 66.67% with $(4, 1, 1)$. Attainable coverage for three alternatives is 66.67% at $K = 2$ and 100% at $K = 3$. Figure 3 reports avoidable loss under both conditions.

The Pooled-AUC Scheme loses coverage in 5/128 original tasks and 5/32 balanced extension tasks at $K = 3$. Table 4's examples were selected post hoc by a deterministic maximum-loss rule. Permutations measure within-task order sensitivity, not additional independent observations.

Table 4: Three task records tracing local relation errors through grouping and selection. Linkage-only comparisons hold scores, threshold, input order, and budget fixed; loss-inducing orders are within-task counts.

Case / configuration	Local error	Observed outcome	Linkage-only outcome	Loss orders
Scheduling Qwen3-P / single, $K = 2$	merge $(0, 2)$	$\{0, 1, 2, 3\}$ $\rightarrow 0, 1$ (B)	$\{0, 1\}$; $\{2, 3\}$ $\rightarrow 0, 2$	8/24
Subset sum Qwen3-P / single, $K = 3$	merge $(1, 2), (2, 4)$	$\{0, 1, 2, 3, 4, 5\}$ $\rightarrow 0, 1, 2$ (C)	$\{0, 1\}$; $\{2, 3\}$; $\{4, 5\} \rightarrow 0, 2, 4$	432/720
Prime numbers TF-IDF / complete, $K = 2$	split	$\{0\}$; $\{1\}$; $\{2\}$; $\{3\} \rightarrow 0, 1$ (B)	unchanged	8/24

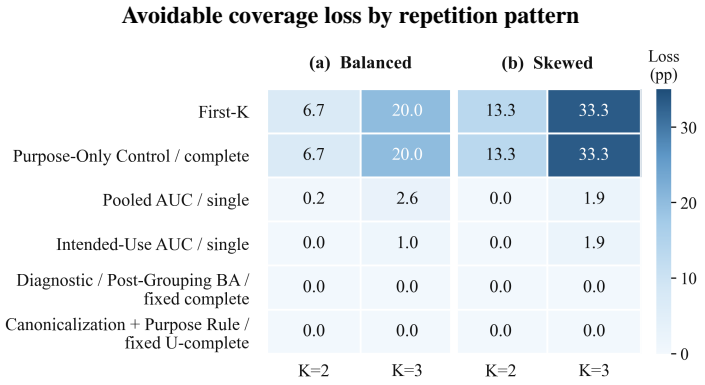


Figure 3: Avoidable Coverage Loss under balanced and imbalanced repetition. Each cell is same-budget attainable coverage minus order-averaged coverage (pp); the 720 orders are averaged within task. Combined labels denote schemes with identical values under these conditions.

4.4 Simple Schemes Can Match Coverage, and Development Ties Matter

The Intended-Use AUC, Intended-Use Post-Grouping BA, Diagnostic, and fixed Qwen3 complete-linkage schemes attain the 100% optimum in the original experiment (Table 2); the evaluated complete-linkage schemes also attain the extension optimum (Table 3). Diagnostic selection thus adds no coverage under these conditions. Development ties can nevertheless conceal different test outcomes: a tied TF-IDF scheme has lower test coverage (Appendix D). Matched-family threshold transfer has small, evaluator-dependent effects (Appendix F.2).

4.5 Pairwise Performance and Functional-Unit Coverage Diverge on Open-Ended Responses

Natural-50 contains 30 final-test queries with six responses each ($K = 3$). Two non-author annotators labeled validity and functional units, with disagreements adjudicated before final test-model results were viewed (Appendices C.6 and F.6). Query Q028 illustrates partial overlap: the Intended-Use AUC Scheme covers 3/6 valid units, whereas another three-response subset covers 4/6.

With Natural-50-development thresholds, single linkage, and frozen response order, Qwen3-U has AUC 0.797 and coverage 71.69%; Qwen3-P has AUC 0.685 and coverage 73.00% (Figure 4, left). AUC uses only valid pairs with unambiguous equivalence labels; partial overlaps, invalid pairs, and uncertain relations are excluded. Coverage instead uses all six candidates. The paired AUC difference is 0.112 (95% CI: 0.025–0.209); coverage differs by 1.30 points in the opposite direction (95% CI: 0–3.91), entirely from one query.

The **Controlled Pooled-AUC Transfer Scheme** uses Qwen3-P/single with its original controlled-development threshold, without Natural-50 refitting. It attains 69.28% mean coverage versus 77.64% mean same-budget attainable coverage, giving 8.36 points of avoidable loss (95% CI: 1.94–16.03) across 6/30 test queries (Figure 4, right). Its mean coverage equals First-K's under the frozen order; the 8.36-point gap is to the attainable optimum, not to First-K.

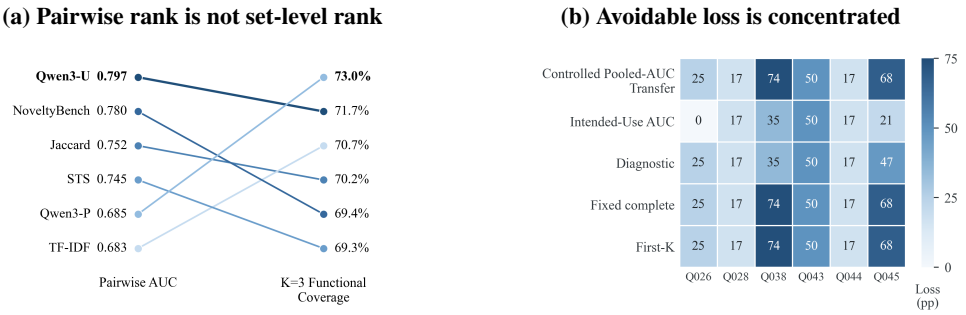


Figure 4: Natural-50 separates pairwise ranking from Set Utility. Left: pairwise AUC versus Functional Coverage under Natural-50 development thresholds, single linkage, and frozen order. Right: Avoidable Coverage Loss (pp) for queries where at least one fixed scheme misses same-budget attainable coverage. Full results are in Appendix F.6.

The Natural-50 Intended-Use AUC Scheme exceeds First-K coverage by 3.72 points (95% paired CI: 0–8.18), while the Diagnostic Scheme achieves 71.29%. Only 12/30 queries permit different coverage across three-response subsets. In a post-hoc analysis of 24 simpler single-unit queries, Qwen3-U exceeds Qwen3-P in AUC by 0.127 (95% CI: 0.028–0.250), while both attain 73.33% coverage. Sensitivity analyses for reference, order, budget, and representative selection appear in Appendix H.

5 Discussion and Limitations

5.1 Converting Pairwise Evidence into Set Utility Requires Three Checks

Budgeted response selection requires three checks. **Intended-Use Alignment** asks whether scores reflect target-use distinctions; **Fixed-Threshold Transfer** asks whether a development-selected threshold remains usable on new sets; and **Selection Compatibility** asks whether grouping and representative selection preserve distinct valid content. Our experiments show failures at each boundary, so pairwise AUC is informative but insufficient for the downstream objective.

Table 5 is diagnostic rather than prescriptive: it localizes where evidence should be inspected, not which intervention must be used. Proposition 1 in Appendix A.3 gives sufficient conditions for attaining the same-budget maximum in the single-object controlled setting, but those conditions assume a correct intended-use reference. In deployment, final adequacy must still be assessed from the selected set.

5.2 Local Errors and Coverage Loss Should Be Evaluated Separately

Incorrect merges can suppress alternatives; incorrect splits can fill slots with duplicates. Input order, repetition, and budget determine whether either error causes loss. Simpler intended-use and fixed complete-linkage schemes match the Diagnostic Scheme in controlled coverage, while Natural-50 coverage rankings change under alternative representative selection (Appendix H). Evaluating the complete pipeline, rather than pairwise accuracy alone, also informs deduplication, retrieval diversification, and candidate reranking.

5.3 Scope of Evidence and Study Limitations

The controlled suite isolates failure mechanisms across four verifiable task families; Natural-50 measures their combined effects on 30 open-ended test queries. Only 12 queries permit different subsets to yield different coverage at $K = 3$, limiting the scope of observed differences. Functional units depend on the predefined use and human annotations, and Natural-50 lacks per-response model identities and generation settings. Its results therefore characterize the evaluated candidate sets rather than deployment-wide failure rates or the isolated effect of any single pipeline stage.

Table 5: Diagnostic checks linking observed failures to stages of the selection pipeline.

Evaluation question	Observed phenomenon → What to inspect	Evidence
Intended-Use Alignment	Strong pooled ranking but weaker intended-use ranking → inspect use composition and the basis of the scores.	Section 4.1: Purpose-Only Control
Fixed-Threshold Transfer	Strong ranking but imbalanced decisions under a fixed threshold → inspect threshold location and decisions on new sets.	Section 4.2: perfect-ranking omissions
Selection Compatibility	Incorrect acceptance across alternatives merges distinct options → compare linkage rules with scores and threshold fixed.	Table 4: scheduling and subset sum
Selection Compatibility	Incorrect splits allow duplicate responses to occupy multiple slots → inspect representative order, slot filling, and truncation.	Table 4: prime numbers
Selection Compatibility	Relation errors occur without reducing coverage → inspect repetition pattern, input order, and selection budget.	Section 4.3: order effects
Selection Compatibility	Responses contain multiple or partially overlapping functional units → inspect the union of valid units retained by the selected set.	Section 4.5: Q028 and post-hoc subsets

6 Conclusion

We introduced a controlled meta-evaluation linking use-conditioned pairwise homogeneity to set utility under a fixed budget. Pooled AUC can reward use cues that do not distinguish target-use alternatives, while perfect intended-use ranking can still lose attainable content after threshold transfer, grouping, and selection. On 30 Natural-50 test queries, the controlled-task Pooled-AUC transfer scheme falls 8.36 percentage points below the same-budget attainable optimum (95% CI: 1.94–16.03), with loss in six queries. Response-selection evaluation should therefore report intended-use ranking, fixed-threshold behavior, Functional Coverage, and Avoidable Coverage Loss.

AI Use Statement

Generative AI was used to assist with research-design discussions, literature searches, figure preparation, and Chinese–English translation. The experimental code was written by the authors, with generative AI providing debugging assistance. Generative AI was not used to produce the human reference labels for Natural-50 or to replace human adjudication. Reference labels for the controlled tasks were determined from task constraints and programmatic verification.

The authors reviewed the citations, scientific claims, and final research materials prepared with AI assistance. All statistics reported in this paper were computed from saved instances, cached scores, and structured results. The authors take full responsibility for the manuscript text, scientific claims, research materials, and results.

Ethics Statement

The initial Natural-50 annotations were produced independently by two non-author annotators with relevant research experience. Disagreements were adjudicated by an author before the final test results were viewed, after which the human reference was frozen. Both original annotation sets were retained for inter-annotator agreement analysis and sensitivity checks. The annotation and adjudication procedures are described in Appendix F.6.

The public data source, response-selection rule, and data splits for Natural-50 are documented in Appendix C.6. Redistribution of third-party materials is subject to their respective licenses. Model weights for which redistribution permission has not been explicitly granted are not included in the released research materials. The scope of the materials and the applicable redistribution restrictions are detailed in Appendix G.

Reproducibility Statement

The anonymous code artifact at https://anonymous.4open.science/r/anonymous-review-code-EC98/ provides the saved instances, evaluator scores, configurations, and scripts needed to recompute the reported analyses without rerunning the evaluator models. Section 3 defines the evaluation and selection procedure; Appendices A and B give the statistical definitions and analysis chronology; Appendix C documents task construction and scoring interfaces; and Appendix F.6 describes the Natural-50 human reference. Appendix G specifies the reproduction commands, included materials, and limits of model rescoring.

References

Jaime Carbonell and Jade Goldstein. The Use of MMR, Diversity-Based Reranking for Reordering Documents and Producing Summaries. Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 335–336., 1998. https://doi.org/10.1145/290941.291025.

Ameet Deshpande, Carlos Jimenez, Howard Chen, Vishvak Murahari, Victoria Graf, Tanmay Rajpurohit, Ashwin Kalyan, Danqi Chen, and Karthik Narasimhan. C-STS: Conditional Semantic Textual Similarity. Proceedings of EMNLP 2023, 5669-5690., 2023. https://doi.org/10.18653/v1/2023.emnlp-main.345.

Dan Friedman and Adji Bousso Dieng. The Vendi Score: A Diversity Evaluation Metric for Machine Learning. Transactions on Machine Learning Research., 2023. https://openreview.net/forum?id=g97OHbQyk1.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. SimCSE: Simple Contrastive Learning of Sentence Embeddings. Proceedings of EMNLP 2021, 6894–6910., 2021. https://doi.org/10.18653/v1/2021.emnlp-main.552.

Yanzhu Guo, Guokan Shang, and Chloé Clavel. Benchmarking Linguistic Diversity of Large Language Models. Transactions of the Association for Computational Linguistics 13, 1507–1526., 2025. https://aclanthology.org/2025.tacl-1.69/.

Shomik Jain, Jack Lanchantin, Maximilian Nickel, Karen Ullrich, Ashia Wilson, and Jamelle Watson-Daniels. LLM Output Homogenization is Task Dependent. arXiv:2509.21267, version 3 revised 2026., 2025. https://doi.org/10.48550/arXiv.2509.21267.

Liwei Jiang, Yuanjun Chai, Margaret Li, Mickel Liu, Raymond Fok, Nouha Dziri, Yulia Tsvetkov, Maarten Sap, and Yejin Choi. Artificial Hivemind: The Open-Ended Homogeneity of Language Models (and Beyond). Advances in Neural Information Processing Systems 38, Datasets and Benchmarks Track., 2025. https://proceedings.neurips.cc/paper_files/paper/2025/hash/754d5a526a5ee5a47220664a0eb92751-Abstract-Datasets_and_Benchmarks_Track.html.

Alex Kulesza and Ben Taskar. Determinantal Point Processes for Machine Learning. Foundations and Trends in Machine Learning 5, 123–286., 2012. https://doi.org/10.1561/2200000044.

Hui Lin and Jeff Bilmes. A Class of Submodular Functions for Document Summarization. Proceedings of ACL-HLT 2011, 510–520., 2011. https://aclanthology.org/P11-1052/.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment. Proceedings of EMNLP 2023, 2511-2522., 2023. https://doi.org/10.18653/v1/2023.emnlp-main.153.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. MAUVE: Measuring the Gap Between Neural Text and Human Text using Divergence Frontiers. Advances in Neural Information Processing Systems 34., 2021. https://arxiv.org/abs/2102.01454.

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. Proceedings of EMNLP-IJCNLP 2019, 3982–3992., 2019. https://doi.org/10.18653/v1/D19-1410.

Ananya B. Sai, Tanay Dixit, Dev Yashpal Sheth, Sreyas Mohan, and Mitesh M Khapra. Perturbation CheckLists for Evaluating NLG Evaluation Metrics. Proceedings of EMNLP 2021, 7219-7234., 2021. https://doi.org/10.18653/v1/2021.emnlp-main.575.

Guy Tevet and Jonathan Berant. Evaluating the Evaluation of Diversity in Natural Language Generation. Proceedings of EACL 2021, 326-346., 2021. https://doi.org/10.18653/v1/2021.eacl-main.25.

Jingxuan Tu, Keer Xu, Liulu Yue, Bingyang Ye, Kyeongmin Rim, and James Pustejovsky. Linguistically Conditioned Semantic Textual Similarity. Proceedings of ACL 2024, 1161-1172., 2024. https://doi.org/10.18653/v1/2024.acl-long.64.

Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Lingpeng Kong, Qi Liu, Tianyu Liu, and Zhifang Sui. Large Language Models are not Fair Evaluators. Proceedings of ACL 2024, 9440-9450., 2024. https://doi.org/10.18653/v1/2024.acl-long.511.

Fan Wu, Emily Black, and Varun Chandrasekaran. Generative Monoculture in Large Language Models. International Conference on Learning Representations, 33068-33107., 2025. https://proceedings.iclr.cc/paper_files/paper/2025/file/5178b2f2d7c44aa390c0777dc77b3f0c-Paper-Conference.pdf.

Ziang Xiao, Susu Zhang, Vivian Lai, and Q. Vera Liao. Evaluating Evaluation Metrics: A Framework for Analyzing NLG Evaluation Metrics using Measurement Theory. Proceedings of EMNLP 2023, 10967-10982., 2023. https://doi.org/10.18653/v1/2023.emnlp-main.676.

Yiming Zhang, Harshita Diddee, Susan Holm, Hanchen Liu, Xinyue Liu, Vinay Samuel, Barry Wang, and Daphne Ippolito. NoveltyBench: Evaluating Language Models for Human-like Diversity. Second Conference on Language Modeling., 2025. https://openreview.net/forum?id=XZm1ekzERf.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. Advances in Neural Information Processing Systems 36, Datasets and Benchmarks Track., 2023. https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

A Statistical Definitions and Finite Counterexamples

A.1 Case-Equal Estimators

Case and relation weights. To prevent cases containing more response pairs from receiving greater weight in the aggregate results, we assign equal total weight to each case containing a relation of a given class, separately for the positive and negative classes. Within each case, this weight is divided equally among the relations belonging to that class. The evaluation unit is a pair of valid responses under a specified use: the positive class indicates that the two responses are equivalent, while the negative class indicates that they are distinct. These labels describe the relation between the responses, not the validity of either individual response.

Let the complete set of cases be $\mathcal{C}$, and let I_c^+ and I_c^- denote the sets of positive and negative relations in case c, respectively. Let $\mathcal{C}^+$ and $\mathcal{C}^-$ denote the sets of cases containing at least one relation of the corresponding class, and let $c(i)$ denote the case to which relation i belongs. The weights of positive and negative relations are

$$w_i^+ = \frac{1}{|\mathcal{C}^+|\,|I_{c(i)}^+|}, \qquad w_j^- = \frac{1}{|\mathcal{C}^-|\,|I_{c(j)}^-|}.$$

The positive and negative classes are normalized separately, so that $\sum_{i:E_i=1} w_i^+ = 1$ and $\sum_{j:E_j=0} w_j^- = 1$. The Equivalent Acceptance Rate, Distinct Rejection Rate, and weighted AUC all use these weights. The weighted AUC compares a relation drawn from the positive class with one drawn from the negative class according to their respective weights: a higher positive-class score contributes 1, a tie contributes $1/2$, and a lower score contributes 0. If either class is absent for a use, its within-use AUC is undefined.

Upper bound on Pooled AUC for a use-invariant scorer. We now derive the 0.750 upper bound used in Section 4.1. Consider a scorer that receives no use information and assigns the same score to a given response pair regardless of use. Let its AUC under the alternative-preservation use be $\mathrm{AUC}_{\mathrm{alt}}$.

Each case contains two distinct alternatives, with two responses per alternative. The four responses form six unordered pairs: two same-object pairs and four different-object pairs. Under the feasibility-confirmation use, all six pairs are labeled equivalent. Under the alternative-preservation use, the two same-object pairs are labeled equivalent, while the four different-object pairs are labeled distinct. Pooling the two uses therefore yields eight positive relations and four negative relations per case. Accordingly, the feasibility-confirmation use contributes $6/8 = 3/4$ of the positive-class weight, and the alternative-preservation use contributes $2/8 = 1/4$. All negative-class weight comes from the alternative-preservation use.

First, consider comparisons between positive relations from the feasibility-confirmation use and negative relations from the alternative-preservation use. Of the positive relations under the feasibility-confirmation use, $1/3$ are same-object pairs, which are precisely the positive relations under the alternative-preservation use. Because the scorer is use-invariant, the ranking performance of this portion against the negative class under the alternative-preservation use is $\mathrm{AUC}_{\mathrm{alt}}$.

The remaining $2/3$ are different-object pairs. They correspond to the same response pairs that form the negative class under the alternative-preservation use and therefore have the same weighted score distribution. When two scores are drawn independently from identical distributions, wins and losses are symmetric, and ties contribute one half. The expected comparison contribution is consequently $1/2$; this does not require every individual comparison to be tied.

Combining the cross-use comparison with the within-alternative-preservation comparison, weighted by the use composition of the positive class, gives

$$\begin{aligned}\mathrm{AUC}_{\mathrm{mixed}} &= \frac{3}{4}\left(\frac{1}{3}\mathrm{AUC}_{\mathrm{alt}} + \frac{2}{3}\times\frac{1}{2}\right) + \frac{1}{4}\mathrm{AUC}_{\mathrm{alt}} \\ &= \frac{1}{4} + \frac{1}{2}\mathrm{AUC}_{\mathrm{alt}} \le \frac{3}{4}.\end{aligned}$$

The $1/3$ and $2/3$ terms inside the parentheses describe the composition of positive relations under the feasibility-confirmation use in terms of same-object and different-object pairs. The outer $3/4$ and $1/4$ terms describe the use composition of the pooled positive class. Qwen3-U achieves an Intended-Use AUC of 1 and therefore attains the 0.750 upper bound on Pooled AUC. This bound depends on the current use composition, relation construction, and weighting scheme; it does not apply to arbitrary pooled-use evaluations.

Use response under fixed responses. For response pairs referring to different canonical objects, the reference relation changes from equivalent under the feasibility-confirmation use to distinct under the alternative-preservation use. Same-object pairs remain equivalent under both uses. We examine these two types of relations separately to assess whether an evaluator makes correct judgments both when a change in intended use should alter the relation and when it should leave the relation unchanged.

Let $\mathcal{F}_c$ denote the four different-object response pairs in case c, and let $\mathcal{H}_c$ denote the two same-object pairs. Let $J_c(x, y)$ indicate whether the judgments for pair (x, y) are correct under both uses: it equals 1 only when both judgments are correct and 0 otherwise. We first compute the average for each relation type within each case and then aggregate across cases:

$$F = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{4} \sum_{(x,y) \in \mathcal{F}_c} J_c(x, y),$$

$$H = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{1}{2} \sum_{(x,y) \in \mathcal{H}_c} J_c(x, y).$$

F measures joint correctness for relations that should change between uses, whereas H measures joint correctness for relations that should remain unchanged. Both require the judgments for the same response pair to be correct under both uses, allowing the two types of use-dependent behavior to be examined separately.

Threshold selection and interval estimation. Thresholds are determined exclusively from the development set. Candidate thresholds include the scores observed on the development set, the midpoints between adjacent observed scores, and the two extreme decisions corresponding to accepting all pairs and rejecting all pairs. We first select thresholds that maximize Balanced Accuracy (BA). If multiple candidates are tied, we retain those that maximize the smaller of the Equivalent Acceptance Rate and Distinct Rejection Rate. If a tie remains, we sort the remaining candidates and choose the lower median: when the number of candidates is even, this is the smaller of the two central candidates. The acceptance rule is consistently defined as $s \geq t$.

For the main-text analyses, bootstrap resampling is performed at the level of complete cases or queries, and paired comparisons reuse the same bootstrap samples. Model scores, development-selected schemes, and thresholds are held fixed when these intervals are computed; only cases or queries are resampled. Sensitivity analyses that refit development thresholds are reported separately in Appendix D for the controlled experiments and Appendix E.3 for the external natural-response pairs.

A.2 Three Counterexamples Separate Ranking, Thresholding, and Aggregation

Perfect ranking does not guarantee that a fixed threshold is usable. For any fixed threshold $t \in (0, 1)$, let every positive-class score equal 1 and every negative-class score equal some $a \in (t, 1)$. Every positive score exceeds every negative score, so AUC is 1. However, both classes lie above the fixed threshold, so every relation is accepted: the Equivalent Acceptance Rate is 1, whereas the Distinct Rejection Rate is 0. This example does not imply that no threshold can separate the two classes. Rather, it shows that perfect ranking does not guarantee that a threshold fixed in advance is correctly positioned for the current score distribution.

BA does not identify which class contains the errors. Let P and R denote the Equivalent Acceptance Rate and Distinct Rejection Rate, respectively. For example, $(P, R) = (1, 0.8)$ and $(0.8, 1)$ both yield a BA of 0.9. In the first case, some distinct response pairs are incorrectly predicted as equivalent; in the second, some equivalent response pairs are incorrectly predicted as distinct. The same average decision performance can therefore correspond to opposite error directions.

More generally, let $m = \min(P, R)$ and $M = \max(P, R)$. Since $2\mathrm{BA} = m + M$ and $M \leq 1$, we obtain $m \geq \max(0, 2\mathrm{BA} - 1)$, and this bound is attainable. BA therefore provides a lower bound on the weaker endpoint, but does not identify which endpoint is weaker or replace separate reporting of the Equivalent Acceptance Rate and Distinct Rejection Rate.

One incorrectly accepted edge can alter many co-cluster relations. Suppose two internally connected clusters A and B are joined by a single erroneous bridging edge. Under single linkage, this edge merges the two clusters, causing all $|A||B|$ cross-cluster response pairs to become co-clustered. The quantity that expands here is the number of post-clustering co-cluster relations, not the number of response pairs directly misclassified by the evaluator.

Complete linkage can block some such bridging merges, but it does not guarantee protection against incorrect splits. In the prime-number case in Table 4, for example, no response pair reaches the threshold. Both linkage rules therefore place every response in a singleton cluster, allowing multiple formulations of the same alternative to occupy separate selection slots. Pairwise errors, grouping outcomes, and final coverage must therefore be evaluated as distinct stages of the pipeline.

A.3 Attainable Coverage and Sufficient Conditions

Attainable coverage under the same budget asks how much distinct valid content can be covered from a fixed candidate set while retaining no more than K responses. A response may contain multiple functional units. For case c, let X_c be the valid candidates used for coverage calculation, and let $H_c(x)$ be the set of functional units covered by response x. When these candidates contain at least one functional unit, attainable coverage is

$$C_c^*(K) = \max_{\substack{S \subseteq X_c \\ |S| \le K}} \frac{\left|\bigcup_{x \in S} H_c(x)\right|}{\left|\bigcup_{x \in X_c} H_c(x)\right|}.$$

The denominator counts the distinct functional units provided by all valid candidates. The maximization considers every response subset satisfying the budget, including subsets that the evaluated selection algorithm does not produce. Repeated occurrences of a unit are counted only once.

In the current controlled experiments, retention is evaluated on the valid-response set

$$X_c^{\mathrm{ret}} = \{x : V_c(x) = 1\},$$

and each response corresponds to exactly one canonical object, so that $H_c(x) = \{a_c(x)\}$. If this set contains M_c distinct canonical objects, selecting at most K responses can cover at most $\min(K, M_c)$ objects. The attainable coverage therefore simplifies to

$$C_c^*(K) = \frac{\min(K, M_c)}{M_c},$$

which corresponds to U_c^* in Equation (4). Invalid responses are used only for the separate Invalid-Response Diagnostic and are not included in this controlled retention set.

Proposition 1 (sufficient conditions for attaining the coverage optimum). In the single-object controlled setting above, suppose that threshold decisions exactly match the object-equivalence relation under the intended use, clustering recovers the corresponding equivalence classes, and the selector follows the rule in Section 3.3 by prioritizing representatives from distinct clusters before filling any remaining slots with duplicate responses. Then the selected set attains the same-budget optimum $C_c^*(K)$.

Proof. Correct threshold decisions and clustering produce M_c clusters, each corresponding to one distinct canonical object. If $K \le M_c$, the selector chooses K representatives from distinct clusters and therefore covers K objects. If $K > M_c$, the selector first chooses one representative from each of the M_c clusters; any additional responses added afterward cannot remove objects that have already been covered. The resulting coverage is therefore

$$\frac{\min(K, M_c)}{M_c} = C_c^*(K).$$

These conditions are sufficient but not necessary. Even when some local relation decisions or grouping steps are incorrect, a particular input order and selection budget may still yield optimal coverage. Moreover, in deployment settings where ground-truth canonical objects are unavailable, the conditions in Proposition 1 cannot be treated as a reliability guarantee that has already been verified.

B Evidence Chronology and Analysis Roles

This appendix records which data are used for each analytical decision, when the corresponding settings are determined, and whether evaluator models are invoked to generate new scores. Table B1 distinguishes controlled designs fixed before performance analysis, interpretive analyses added after results became available, new-set evaluations frozen before scoring, and diagnostics conducted on external sources. Internal freezing records the implementation chronology within the research group and should not be interpreted as external preregistration.

Table B1: Evidence chronology and analysis roles. The table records which data support each analysis, when the corresponding choices were fixed, and whether evaluator models were invoked to generate new scores. "Internally fixed/frozen" describes implementation chronology within the study and does not imply external preregistration.

Analysis	Data use / decision rule	Timing and model scoring
Controlled relation evaluation	64 development cases select thresholds and operating schemes; 128 test cases are used only for evaluation.	**Pre-analysis fixed** New model scores: **Yes**
Purpose-Only Control; Canonicalization + Purpose Rule; use-response analysis	Reuse controlled instances, existing scores, and fixed thresholds.	**Post-hoc interpretive** New model scores: **No**
24 response orders and use composition	Reuse the same tasks and scores; rerun grouping for every permutation.	**Post-hoc interpretive** New model scores: **No**
Three-alternative extension	Freeze generation rules, threshold-transfer procedure, candidates, budgets, and statistics before scoring.	**Pre-score frozen** New model scores: **Yes**
Per-task loss chains	Reuse permutation records; select illustrative cases using a fixed rule.	**Post-hoc interpretive** New model scores: **No**
Development resampling and tied schemes	Reuse development fits and test records.	**Post-hoc robustness** New model scores: **No**
External natural response pairs	Use public relation labels; natural development data are used only for threshold adaptation.	**External transfer diagnostic** New model scores: **Yes**
Natural-50 complete sets	Freeze the human reference first; use 10/9/30 eligible queries for threshold development, scheme selection, and final testing.	**Pre-score frozen** New model scores: **Yes**

The research hypothesis for the three-alternative experiment was proposed after the controlled main results had become available. However, its generation rule and evaluation protocol were fixed before evaluator models were run on the new test set. The source test set used for the external natural response-pair analysis had already been used in the original study from which it was obtained; we therefore use it only as an external-scope diagnostic. The exact source is described in Appendix E.

The Natural-50 human reference was frozen before model results for that dataset were viewed. The original plan allocated 10, 10, and 30 queries to threshold development, scheme selection, and final testing, respectively. After Q012 was excluded in advance, the realized split became 10, 9, and 30. Physically separate score files were used during development, and the main-analysis final test results were revealed only after byte-level verification of the development selection. The corresponding frozen records are documented in Appendix G.

Before Natural-50 model scoring, the Qwen input limit was increased from 1,024 to 2,048 tokens to preserve complete natural responses. This interface revision, together with two other implementation revisions, was timestamped before scoring and did not alter labels, selection budgets, candidate sets, or reported statistics.

C Controlled Tasks, Scoring Interfaces, and Selection Protocol

This appendix documents the construction of the controlled tasks, the scoring interfaces, and the selection protocols used for both the controlled suite and Natural-50. The main text evaluates response validity, use-conditioned pairwise relations, and set-level coverage under a fixed selection budget as separate stages. Here, we specify how references are constructed at each stage, what information is available to the evaluators, how thresholds and operating schemes are selected from development data, and the currently verifiable source boundary of Natural-50.

Table C1: Equal response validity does not imply the same functional relation. Each row first determines response validity using an independent task-constraint check and then applies the stated intended use to define the relation between valid responses.

Response pair	Both responses valid?	Intended use	Reference relation
`23` and `23.0`; alternative formulations of the same object	Yes	Alternative-Preservation Use	**Equivalent**; formatting variation does not create a new alternative
`23` and `29`; two different feasible solutions	Yes	Feasibility-Confirmation Use	**Equivalent**; either valid witness is sufficient for the use
The same pair, `23` and `29`; two different feasible solutions	Yes	Alternative-Preservation Use	**Distinct**; the alternatives should be preserved separately
One valid response and one out-of-range, illegal, or non-optimal response	No	Either explicit use	Excluded from valid–valid relation evaluation; handled by the Invalid-Response Diagnostic
Two valid but different responses	Yes	Only states "provide a useful response"	Intended use is ambiguous; no binary relation label is assigned

C.1 Use Determines Functional Relations Between Valid Responses

The controlled tasks first determine whether each individual response satisfies the task constraints and then define the functional relation between two valid responses according to the intended use. For case c and response x, let $V_c(x)$ indicate whether the response is valid, and let $a_c(x)$ denote its parsed canonical object. For two valid responses, the Feasibility-Confirmation Use treats any pair of valid responses as equivalent, whereas the Alternative-Preservation Use labels them equivalent only when they refer to the same canonical object. Integers, decimals, and other equivalent representations are canonicalized to the same object, while different valid numerical values, subsets, paths, or time intervals remain distinct.

If the intended use is ambiguous, no binary functional-relation label is assigned. Response pairs containing an invalid response are likewise excluded from the valid–valid relation evaluation and are instead handled by the separate Invalid-Response Diagnostic. Table C1 illustrates these cases using fixed examples. The examples are included only to explain reference construction; the actual tasks and responses are stored in the frozen experimental data.

Validity constraints differ across the four task families. Prime-number tasks require the answer to be prime and strictly within the specified open interval. Subset-sum tasks require unique members whose values sum exactly to the target. Shortest-path tasks require every directed edge to be legal and the complete path to attain the global optimum. Scheduling tasks require integer-valued times, the specified duration, and placement within the common availability interval.

For each response, the parser extracts only the answer object, and the constraint validator uses that object to produce the validity and canonical-object records. Evaluators receive only the natural-language task, the intended-use description when permitted by the scoring interface, and the response text. They do not receive validator outputs, validity labels, reference relations, task-family fields, or case identifiers.

Across the controlled test set, the two intended uses together produce 12 valid–valid relation judgments per case: 8 equivalent relations and 4 distinct relations. An additional 4 relations containing invalid responses are reserved for the Invalid-Response Diagnostic, and 1 relation with ambiguous intended use remains unlabeled. The 128 test cases therefore yield 1,536 valid–valid relation judgments, 512 invalid-response relations, and 128 ambiguous relations. The independent statistical units remain the 128 complete cases rather than the individual response pairs.

Under the Feasibility-Confirmation Use, all valid response pairs are equivalent. This use therefore contains no negative class, and its standalone AUC is undefined. Development thresholds are

Table C2: Scoring models, components, and fixed input settings.

Model or component	Role and fixed input setting
`yimingzhang/deberta-v3-large-generation-similarity` (NoveltyBench)	Functional-equivalence classifier; each response is truncated to its first 128 tokens.
`microsoft/deberta-v3-large` tokenizer	Tokenizer specified by the official NoveltyBench classification code.
`Qwen/Qwen3-4B-Instruct-2507` (Qwen3-P / Qwen3-U)	Both interfaces use the same quantized model weights; Qwen3-P receives task and intended-use information, whereas Qwen3-U receives neither; only English inputs are evaluated.
`cross-encoder/stsb-roberta-base` (STS)	Saved quantized ONNX interface with a bidirectional sigmoid readout.

fitted using only valid–valid relations, keeping functional-equivalence decisions separate from the Invalid-Response Diagnostic.

C.2 Fixed Models, Input Interfaces, and Parsing Rules

Table C2 lists the primary scoring models, tokenizers, and input settings used in this study. The models were chosen based on task relevance, public availability, and local computational resources rather than performance on the new test sets. Complete version identifiers and licensing records for all components are included with the recomputation materials described in Appendix G.

The NoveltyBench interface uses the official pretrained classifier and interprets the probability assigned to class 1 as the functional-equivalence score. We do not use the original decision threshold, data split, or deduplication procedure from NoveltyBench. Instead, thresholds are re-estimated on our own development set, and all pairwise and set-retention evaluations are conducted under the common protocol defined in this paper. The original project code is released under the MIT license, whereas the model card for the fine-tuned weights does not explicitly specify a redistribution license; these two facts are recorded separately in the released documentation.

The two Qwen interfaces share the same model weights and numerical readout. Qwen3-P receives a fixed system instruction together with the current task and intended-use description. Qwen3-U receives the same system instruction but omits both the task and intended-use information. Scoring uses only the next-token logits for `Yes` and `No` to compute a conditional probability. For each evaluation, we additionally store the most likely token, raw generated text, both logits, input length, and response order.

The prompts contain no reference labels, role markers, validator outputs, task-family fields, or case identifiers. When the two-class `Yes/No` score can be computed, the resulting continuous score is retained even if the generated natural-language output does not fully comply with the requested format. Format compliance is reported separately and is not conflated with correctness of the functional-relation judgment.

The STS and NoveltyBench interfaces are evaluated using their fixed truncation and parsing rules, with no adjustment based on test-set performance. Truncation behavior, parsing failures, and response-order sensitivity on the external natural-response pairs are reported separately in Appendix E.

C.3 Development Data Determine Thresholds and Operating Schemes

Continuous equivalence scores must first be converted into binary relation decisions and then passed through clustering and representative selection. All operating thresholds are determined exclusively from development data and held fixed at test time. For the controlled suite, candidate thresholds include scores observed on the development set, midpoints between adjacent unique scores, and the two extreme decisions corresponding to accepting all pairs and rejecting all pairs. Threshold selection first maximizes Balanced Accuracy (BA). If multiple candidates are tied, we retain those

that maximize the smaller of the Equivalent Acceptance Rate and Distinct Rejection Rate. Any remaining tie is resolved by sorting the candidate thresholds and selecting the lower median. The acceptance rule is uniformly defined as $s \geq t$.

Under the Alternative-Preservation Use, we further evaluate the consequences of passing relation decisions into set selection. The incorrect-merge rate is the proportion of truly distinct response pairs that are placed in the same predicted cluster, whereas the incorrect-split rate is the proportion of truly equivalent response pairs that are placed in different predicted clusters. In the original four-response setting, each set contains four valid responses representing two distinct canonical objects, and the selection budget is fixed at $K = 2$.

Representative selection follows a deterministic input-order rule. Predicted clusters are first ordered by the earliest response position appearing in each cluster, and the earliest response within each cluster is chosen as its representative. If the number of cluster representatives exceeds the budget, only the first K representatives are retained. If fewer than K representatives are available, unselected responses are added in their original input order until the budget is filled. The output therefore always contains exactly K responses. In the controlled setting, Functional Coverage measures only the distinct valid canonical objects retained; it does not evaluate linguistic quality or complete semantic clustering quality.

Operating schemes are also selected exclusively from development data. The Pooled-AUC Scheme selects the scoring interface using Pooled AUC and applies prespecified single linkage. The Intended-Use AUC Scheme likewise selects the scoring interface using Intended-Use AUC and applies prespecified single linkage. The Diagnostic Scheme jointly compares scoring interfaces and linkage rules. For each task family, it computes the Equivalent Acceptance Rate and Distinct Rejection Rate from the post-grouping co-cluster relation and first maximizes the minimum of these two decision rates across all families. Ties are then broken by development Functional Coverage, followed by Post-Grouping BA, and finally by deterministic sorting of model and linkage names. All selection criteria are computed from the development set; test results are never used for scheme selection.

For the main paired interval estimates, the development-selected schemes and thresholds are held fixed. Paired comparisons use the same set of 1,000 family-stratified bootstrap resamples of complete cases. All prespecified comparisons are reported, including zero and negative differences. Sensitivity to variation in the development sample and to threshold refitting is analyzed separately in Appendix D.

C.4 Chronology of Technical Revisions and Interpretive Controls

Before performance analysis, the controlled suite internally fixed the data, candidate scoring interfaces, threshold-selection rule, and reported statistics. Two implementation details were subsequently identified and corrected. The first unified the canonical-object equality rule for integer and decimal representations. The second added an integer-minute constraint to the scheduling tasks. The validity judgments for all 960 fixed responses were unchanged before and after the latter revision, and the relation labels, scoring interfaces, and stopping rules were also unchanged. These records document the implementation chronology within the research group and do not constitute external preregistration.

The Purpose-Only Control, Canonicalization + Purpose Rule, and full use-response analysis reported in the main text were added after the primary results had become available and are therefore treated as post-hoc interpretive controls. They reuse existing instances, cached scores, and fixed thresholds, do not require additional evaluator-model invocations, and did not participate in the original scheme-selection process.

The three-alternative extension follows a different chronology. Its research hypothesis was motivated by the controlled main results, but the generation rules for the new sets, candidate responses, selection budgets, threshold-transfer procedure, and reported statistics were fixed before evaluator models were run on that new test set. The complete analysis chronology and model-invocation status are summarized in Appendix B.

Table C3: Original task text for the same response pair under two intended uses. `q1` is used for feasibility confirmation and `q2` for alternative preservation; they are never provided together in a single evaluation.

Input field	Saved original English text
Response `x`	I propose the following response: {"answer":73}. This is my answer.
Response `y`	Here is the response I propose. {"answer":89} End of response.
`q1`: feasibility confirmation	Return one prime number strictly between 67 and 167. Purpose: only certify that at least one feasible witness exists. The exact choice of valid witness is discarded. Any two valid witnesses are functionally interchangeable; an invalid response is not interchangeable with a valid one.
`q2`: alternative preservation	Return one prime number strictly between 67 and 167. Purpose: preserve distinct feasible answers for a user choosing between alternatives. Two valid responses are functionally interchangeable only if they specify the same concrete number, subset, path, or time interval. Ignore wording, formatting and integer versus decimal notation. An invalid response is not interchangeable with a valid one.

C.5 Scorer-Visible Inputs and Evaluation-Only References

Table C3 presents a fixed case from the development set. The two responses provide the values 73 and 89, while the underlying task asks for a prime number strictly between 67 and 167. The same response pair is evaluated under two intended uses: the Feasibility-Confirmation Use requires only that a feasible answer be provided, whereas the Alternative-Preservation Use requires distinct valid alternatives to remain distinguishable. The case is shown in its frozen response order, and the table preserves the original English text stored in `interface.jsonl`.

In addition to the fixed system instruction, Qwen3-P receives the task text corresponding to the current intended use (`q1` or `q2`) together with responses `x` and `y`. Qwen3-U receives the same system instruction and the same response texts but receives neither the task nor the intended-use description.

Evaluation-only reference. Independent constraint validation confirms that both responses are valid, with canonical objects 73 and 89. Under the Feasibility-Confirmation Use, either response satisfies the confirmation objective, so the reference relation is equivalent (`expected=1`). Under the Alternative-Preservation Use, the two responses provide different valid options, so the reference relation is distinct (`expected=0`). These reference values are used only to evaluate scorer judgments and are never provided to the scorer.

Task constraints and reference relations are stored separately in `cases.jsonl` and `relations.jsonl`, apart from the scorer-input files. The intended-use description therefore provides a general criterion for judging functional equivalence rather than revealing the correct label for the particular response pair being evaluated.

C.6 Natural-50 Data Composition and Evaluation Setup

Natural-50 is derived from the human absolute-rating dataset released with Infinity-Chat (Jiang et al., 2025), identified in the project records as liweijiang/infinite-chats-human-absolute at public commit `b5134adae8b0e9879c32b7f7d1eb1f9526833d95`. The source `train` split contains 50 queries with 15 responses per query. The released artifact contains the 300 selected responses in `data/current_natural50/source_responses.csv`; it does not contain the upstream Parquet file.

We retain all 50 source queries. Within each query, a SHA-256 sorting key is computed from the fixed seed `infinity50-structure-v1-20260825`, the query identifier, and the response identifier. The six responses with the smallest keys are selected, yielding 300 responses. This rule does not use the source human ratings. The released CSV preserves the selected response texts, identifiers, ranks, and sorting keys. Because the release does not include the complete upstream file or a source-selection verification record, the upstream file identity and the six-of-fifteen selection cannot be independently checked from this artifact alone. The selected records do not establish individual

response-model identities or generation settings, so we do not attribute responses to specific models or settings.

The source responses are mapped through a fixed correspondence to the blinded annotation package and analysis inputs. Within each query, the six responses are stored in a frozen A1–A6 order. We separately annotate intended use, response validity, and the functional units contained in each response. These annotations are distinct from the human ratings included in the source dataset. The human reference is frozen before final-test evaluator results are viewed; the annotation and adjudication procedures are described in Appendix F.6.

Q012 asks which five-letter English word is "most hot," without specifying the intended meaning of "hot" or the criterion by which candidate words should be compared. After human review, its intended-use applicability remained uncertain. Under an applicability rule fixed before model results were revealed, Q012 was therefore excluded from the scheme-selection development split. After this exclusion, 10, 9, and 30 queries are used for threshold development, scheme selection, and final testing, respectively. The final test set is unchanged. Thresholds and operating schemes are determined from the corresponding development data and then held fixed for final testing.

The main analysis treats the frozen six-response sets as the evaluation objects. For each query, three of the six responses are retained, and Functional Coverage is computed from the valid functional units contained in the selected responses. To obtain attainable coverage under the same budget, we enumerate all response subsets from the same candidate set that satisfy the same selection budget. The gap between the observed coverage and this same-budget optimum is the Avoidable Coverage Loss: it captures omissions that could have been avoided by a different selection from the same candidates, while excluding coverage limitations imposed solely by the selection budget.

The frozen human reference and saved continuous scores support offline recomputation of grouping, representative selection, Functional Coverage, and Avoidable Coverage Loss for the released selected-response sets. The current artifact does not include the original source Parquet file or the separate source-recovery verification records.

D Sensitivity of Development Selection to Resampling and Ties

This post-hoc sensitivity analysis, conducted after the main results became available, examines whether variation in the development sample changes the selected scheme or its coverage performance. We also compare the test performance of schemes that are tied under the development criterion.

Resampling setup. Within each of the four task families, complete cases are sampled with replacement, with 1,000 resamples for each setting. We distinguish three sources of variation:

1. **Test variation only:** The scheme and threshold selected in the original development phase are held fixed, and only the test cases are resampled.
2. **Development variation only:** Development cases are resampled and thresholds are refitted, after which performance is evaluated on the complete test set.
3. **Joint development and test variation:** Development cases are resampled and thresholds are refitted while test cases are also resampled.

For each setting, sensitivity ranges are summarized by the empirical 2.5th and 97.5th percentiles. Within each task, Functional Coverage is first averaged over all 24 or 720 response orders before differences between schemes are computed. Paired scheme comparisons use the same resamples under the corresponding repetition condition.

Scheme selection and coverage differences. Across all 1,000 development resamples, the Pooled-AUC Scheme always selects Qwen3-P with single linkage, whereas the Diagnostic Scheme always selects Qwen3-P with complete linkage. The fitted development thresholds, however, vary across resamples. Table D1 compares the order-averaged Functional Coverage of the Diagnostic Scheme with that of the Pooled-AUC Scheme.

This sensitivity analysis uses a separately saved set of test-resampling records. Consequently, the upper endpoint of the test-only range for the original four-response condition is 1.17 percentage

Table D1: Sensitivity of order-averaged Functional Coverage differences to development and test resampling (percentage points). Each difference is the Diagnostic Scheme minus the Pooled-AUC Scheme; positive values therefore indicate higher coverage for the Diagnostic Scheme. The point estimate corresponds to the main-text result, and the final three columns report empirical 2.5th–97.5th percentile ranges under the three resampling settings.

Set condition	Point estimate	Test variation only	Development variation only	Joint development and test variation
Original four responses, $K = 2$	0.65	[0.13, 1.17]	[0.52, 1.56]	[0.13, 1.95]
Balanced three-alternative set, $K = 3$	2.57	[0.63, 4.65]	[2.43, 4.17]	[0.97, 5.28]
Imbalanced three-alternative set, $K = 3$	1.88	[0.00, 3.75]	[1.88, 2.08]	[0.00, 3.96]

points, rather than the 1.18 percentage points reported from the main-text resampling record. The point estimate and the comparison itself are unchanged.

For the original four-response condition and the balanced three-alternative condition, all three resampling ranges remain above zero, preserving the direction of the corresponding main-text comparisons. For the imbalanced three-alternative condition, the joint development-and-test range is [0.00, 3.96] percentage points and includes zero. In this condition, 317 of the 1,000 refitted development thresholds for the Diagnostic Scheme do not attain 100% coverage. Thus, although development resampling does not change the selected scoring interface or linkage rule, variation in the fitted threshold can still affect final coverage.

Test performance of development-tied schemes. Before the final deterministic name-based tie break, five schemes are tied under the diagnostic development criterion. Under the original fixed response order, their test coverage ranges from 89.84% to 100%. The lower values are produced by the two TF-IDF linkage schemes, whereas the schemes attaining 100% include Qwen3-P with complete linkage.

These fixed-order results are reported separately from the all-order averages in Table D1. Deterministic name-based sorting identifies the final selected scheme, but a tie on the development criterion does not imply equal test coverage among the tied schemes. This analysis is diagnostic only: test results are not used to reselect an operating scheme.

E External Natural-Relation Reference

This appendix uses publicly available natural response pairs to examine functional-relation ranking and threshold transfer outside the controlled tasks. We compare relation decisions obtained with the controlled-development thresholds and with thresholds fitted on natural development data, and then examine input truncation, response order, and probability readout. This analysis concerns pairwise functional relations only; it does not provide set-level relation annotations or Functional Coverage labels for Natural-50.

E.1 Public Relation Reference and Evaluation Setup

The data are drawn from the fixed-version `train/val` files in the official NoveltyBench repository. The complete version identifier is preserved with the recomputation materials described in Appendix G. The original authors annotated functional relations for 1,100 response pairs, using 1,000 pairs for classifier training and 100 for evaluation; the latter were also used in the original paper to compare models and run settings (Zhang et al., 2025). We use these public data only for a post-hoc transfer diagnostic. We do not use the automatic grouping or quality scores in `humans.jsonl`; although that file contains responses written by the original authors, it does not provide a directly usable human reference for complete clustering.

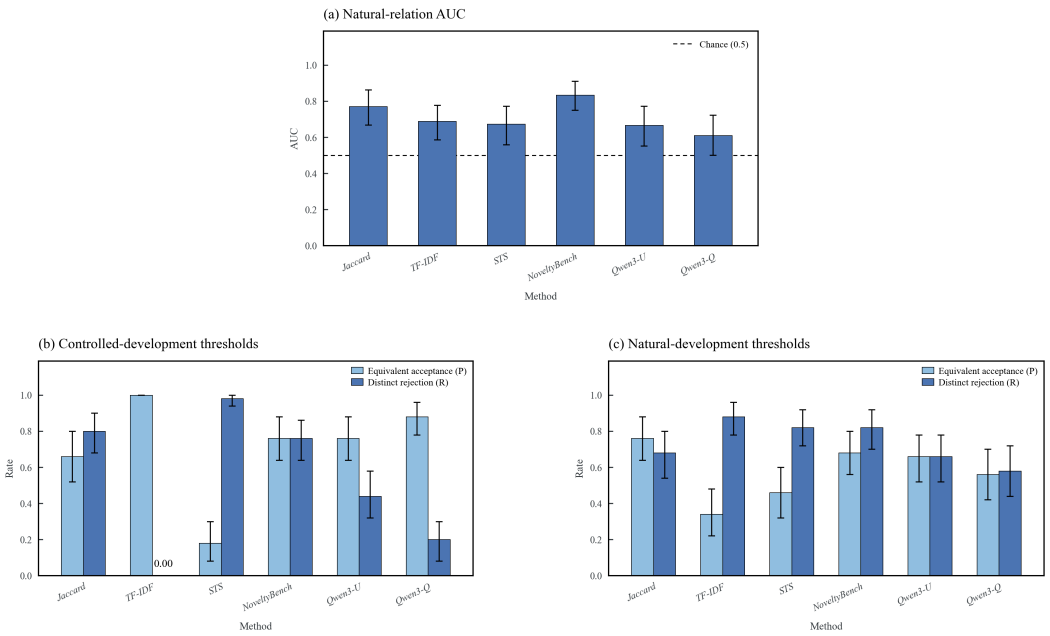


Figure S1: Ranking and threshold transfer on the natural-relation reference. Qwen3-Q and Qwen3-U denote the query-conditioned and task-agnostic interfaces, respectively. All three panels use the same 100 validation queries, with the six scoring interfaces on the x-axis and the corresponding metric on the y-axis. Panel A reports Natural-relation AUC for each interface, with a horizontal dashed line at the random-ranking level of 0.5. Panels B and C use the controlled-development thresholds and natural-development thresholds, respectively. Within each group, the left bar shows the Equivalent Acceptance Rate P and the right bar shows the Distinct Rejection Rate R. Error bars are the existing 95% query-bootstrap intervals. In Panels B and C, the corresponding thresholds are held fixed while only the validation queries are resampled, with a denominator of 50 for each decision rate. In Panel B, TF-IDF has a Distinct Rejection Rate of 0 and is therefore shown with a zero-value label.

The validation file contains 100 distinct queries and their corresponding response pairs, with 50 labeled equivalent and 50 labeled distinct. Every `labeler` field identifies the same annotator. We find no overlap between the validation file and the source training file when comparing task identifiers, task texts normalized for whitespace and case, and exactly identical unordered response pairs.

To construct a natural development set, we select 100 distinct queries from the source training file using a fixed hash order without consulting the labels during selection. The resulting development set contains 30 equivalent and 70 distinct response pairs. Because these development examples were also part of the data used to train the NoveltyBench classifier, that classifier's performance on this development set is not treated as new independent validation evidence.

We denote the query-conditioned interface by Qwen3-Q and the task-agnostic interface by Qwen3-U. Under a generic functional-interchangeability instruction, Qwen3-Q receives the original source query but does not receive the Feasibility-Confirmation Use or Alternative-Preservation Use descriptions from the controlled suite. Qwen3-U omits the source query as well. Source model identities, annotator identities, and relation labels are never provided as scoring inputs.

The source labels describe functional relations between response pairs but do not separately record response validity, annotation disagreement, confidence, or multi-annotator agreement. We therefore evaluate agreement with the source relation labels without interpreting every `distinct` label as evidence that a response is invalid, and without extrapolating these pairwise labels to coverage in multi-response sets. The overlap checks above also do not rule out cross-task near duplicates, model-pretraining contamination, or subjectivity in the reference.

The NoveltyBench classifier shows strong relation-ranking performance on this source, whereas Qwen3-Q under the controlled-development threshold is strongly biased toward accepting equivalence. Thresholds fitted on the natural development set change the tradeoff between the two decision rates;

Table E1: Thresholds and input records for natural-relation evaluation. The final two columns both use the 100 validation queries as the denominator. Order disagreement means that, after swapping the two responses, the separate decisions in the two directions differ under the original controlled threshold; the final score remains the mean over both directions. No parsing failure was recorded for any of the six interfaces.

Interface	Controlled development threshold	Natural development threshold	Truncated queries	Order disagreement queries
Jaccard	0.320512821	0.275916454	0	0
TF-IDF	0.391268148	0.972031929	0	0
STS	0.727340102	0.663592339	75	6
NoveltyBench classifier	0.134904414	0.361143000	88	4
Qwen3-U	0.749566982	0.999962121	0	12
Qwen3-Q	0.497087482	0.999999911	0	6

the adapted BA values and their changes are reported in Table E2. Figure S1 retains the original probability readout, while a post-hoc diagnostic of score saturation is reported in Appendix E.4. The interfaces differ in supervision source, truncation setting, and access to task information, and these differences are documented explicitly in the analyses below.

E.2 Thresholds, Input Truncation, and Response Order

Table E1 reports both threshold sets together with the numbers of queries affected by input truncation and response-order disagreement. All computations use the full-precision thresholds stored in JSON; the decimal values shown in the table are for presentation only and are not substituted for the stored values during evaluation.

Both Qwen interfaces successfully produce `Yes/No` outputs in both response orders. Under the controlled-development thresholds, however, Qwen3-U and Qwen3-Q yield different binary decisions across the two directions for 12 and 6 queries, respectively. Output-format compliance, directional consistency, and correctness of the functional-relation judgment are distinct evaluation properties. Bidirectional averaging is part of the fixed scoring rule, and the two response orders are not treated as independent observations.

STS and NoveltyBench retain their prespecified truncation rules, with no adjustment to input length based on validation performance. Under these rules, 75 of the 100 validation queries are affected by STS truncation and 88 by NoveltyBench truncation. The present analysis does not isolate the causal contribution of truncation from the underlying semantic capability of either interface.

TF-IDF retains the vocabulary and inverse-document-frequency weights fitted on the 304 unique development responses from the controlled suite; neither the vocabulary nor the IDF weights are refitted on the natural data. Under the controlled-development threshold, TF-IDF predicts all validation response pairs as equivalent. Natural-development adaptation changes only the decision threshold and the resulting binary decisions, not the fitted TF-IDF representation.

Two TF-IDF cosine scores slightly exceeded 1 because of floating-point rounding. A separately recorded numerical correction clips values that fall outside the valid range only within numerical tolerance back to $[0, 1]$, while preserving the original values for auditability. The original fixed code, reference labels, and statistical procedures are not overwritten.

E.3 Adaptation Performance of Natural-Development Thresholds

We jointly examine sampling variation in the development threshold and in the validation queries. In each resampling repeat, we sample with replacement within the 30 equivalent and 70 distinct queries of the natural development set, and independently sample with replacement within the 50 equivalent and 50 distinct queries of the validation set. Development and validation sampling use independent

Table E2: Adapted BA and refitting sensitivity under natural development. Intervals are empirical 95% percentile ranges from 1,000 resamples. Every BA difference is defined as BA under the natural-development threshold minus BA under the original controlled threshold and is reported on the proportion scale. Paired differences use the same validation queries within each repeat. Repetition does not increase the number of independent queries.

Interface	Adapted BA point estimate	Refitted BA interval	BA-difference point estimate [interval]
Jaccard	0.720	[0.600, 0.810]	-0.010 [-0.130, 0.040]
TF-IDF	0.610	[0.530, 0.710]	0.110 [0.030, 0.210]
STS	0.640	[0.500, 0.710]	0.060 [-0.090, 0.130]
NoveltyBench classifier	0.750	[0.670, 0.830]	-0.010 [-0.060, 0.040]
Qwen3-U	0.660	[0.540, 0.760]	0.060 [-0.030, 0.130]
Qwen3-Q	0.570	[0.470, 0.690]	0.030 [-0.060, 0.140]

random streams. Within each repeat, however, all scoring interfaces and both threshold settings share the same resampled validation queries so that method comparisons remain paired at the query level.

Development-query multiplicities are incorporated as weights during threshold selection. Thresholds are still selected by first maximizing Balanced Accuracy (BA), then maximizing the lower of the Equivalent Acceptance Rate and Distinct Rejection Rate, and finally applying the median-member tie-breaking rule. Models, prompts, and TF-IDF features are held fixed throughout this analysis. The procedure preserves the observed class counts and does not retrain any model or simulate uncertainty in the reference labels.

For TF-IDF, BA increases by 0.110, with a paired difference interval of [0.030, 0.210]. Its adapted Equivalent Acceptance Rate and Distinct Rejection Rate are 0.340 and 0.880, respectively, indicating that the two decision rates remain substantially imbalanced despite the increase in BA.

For each of the other five interfaces, the BA-difference interval includes zero. The corresponding point estimates include both positive and negative changes across interfaces. Natural-development threshold adaptation therefore does not produce a consistent improvement across scoring interfaces.

As a direct comparison between the two Qwen interfaces under the original probability readout, the AUC difference for Qwen3-Q minus Qwen3-U is -0.0568, with a 95% paired interval of [-0.1150, -0.0066]. This comparison describes the ranking difference under the current interfaces and source reference and is not adjusted for multiple comparisons.

E.4 Probability Saturation and Stable Numerical Readout

The natural-development thresholds for the two Qwen interfaces are extremely close to 1. After obtaining the main results, we therefore examine whether FP32 probability rounding causes otherwise distinct scores to collapse into ties. For the two response orders $k = 1, 2$, let d_k denote the cached difference between the `Yes` and `No` logits, and define $\sigma(d) = 1/(1 + e^{-d})$. The original score is the arithmetic mean of the `Yes` probabilities in the two directions,

$$\bar{p} = \frac{\sigma(d_1) + \sigma(d_2)}{2}.$$

The corresponding log odds can be written as

$$z = \log \frac{\bar{p}}{1 - \bar{p}} = \log \left[\sigma(d_1) + \sigma(d_2)\right] - \log \left[\sigma(-d_1) + \sigma(-d_2)\right].$$

Under exact arithmetic, z is a monotonic transformation of the original bidirectional probability mean. The implementation computes z directly from the cached logits using numerically stable `log-sigmoid` and `logaddexp` operations, rather than first forming probabilities that may round

Table E3: Stable ranking diagnostic from the same cached logits. All results use the same 100 validation queries. The stable readout is used only to recompute AUC and does not produce a new set of threshold decisions.

Interface	Original score exactly 1	Original-probability AUC	Stable-log-odds AUC interval
Qwen3-U	37/100	0.666	0.656 [0.544, 0.763]
Qwen3-Q	44/100	0.610	0.627 [0.513, 0.739]

to values extremely close to 0 or 1. This stable computation can therefore distinguish scores that become tied under the original probability readout.

This analysis is a post-hoc numerical diagnostic. It does not use reference labels to construct the transformation, fit any parameters, invoke the evaluator models again, or reselect thresholds. It only recomputes AUC from the same cached logits using a numerically stable readout and does not replace the operating setting used in the main analysis.

The stable readout increases the number of distinct scores for Qwen3-U and Qwen3-Q from 62 and 49 to 100, respectively, removing numerical ties in the current validation scores. The AUC change for Qwen3-Q is 0.0176 with a 95% interval of [-0.0174, 0.0530], while the change for Qwen3-U is -0.0108 with an interval of [-0.0360, 0.0120]. Both change intervals include zero, so the increased numerical resolution does not correspond to a clear AUC improvement. This check targets output-probability rounding and does not isolate effects of weight quantization, internal numerical precision, task interpretation, or the source reference.

This diagnostic specifically targets rounding in the output-probability readout. It does not isolate possible effects of weight quantization, internal numerical precision, task interpretation, or properties of the source reference.

E.5 Disagreement Cases Between Source Labels and Model Judgments

Table E4 presents four validation cases in which Qwen3-Q, evaluated with the original threshold of 0.497087482, disagrees with the source label. These examples illustrate distinctions involving concrete alternatives, information content, and the criteria used to define functional equivalence. They are included to interpret existing disagreements rather than to estimate the prevalence of particular disagreement types, and they do not alter the source labels.

In the first two cases, responses with similar themes or templates contain different concrete options or additional reasons, yet both pairs are predicted equivalent under the controlled-development threshold. After switching to the natural-development threshold, the truffle/cheese case is predicted distinct, whereas the Vue.js case remains accepted as equivalent. These examples show that threshold adaptation can alter some individual decisions without eliminating all disagreements with the source labels. Unlike the controlled tasks, the source queries generally do not explicitly distinguish between the Feasibility-Confirmation Use and the Alternative-Preservation Use.

The latter two cases illustrate differences in how functional equivalence may be interpreted by the source reference. The two jokes contain different content but are labeled equivalent, indicating that the source relation should not be interpreted simply as canonical-object identity. In the password case, the responses differ on a directly verifiable length requirement, but the source data provide no separate response-validity annotation. Disagreement with the source relation therefore cannot be interpreted uniformly as either an evaluator error or a loss of valid information.

These interpretations are based only on the existing source texts, source labels, and saved model judgments. They are not new independent human annotations and do not relabel the source data. The complete tasks, responses, anonymous identifiers, and predictions under both threshold settings are retained in the error-analysis materials.

Table E4: Content summaries and model judgments for cases in which Qwen3-Q disagrees with the source label. Row numbers are zero-based indices from the source `val.jsonl` file, and every `labeler` field is `Yiming`. Source label 0 denotes distinct and source label 1 denotes equivalent. Model scores are the original bidirectional probability means. Response contents are summarized by the authors from the saved original texts.

Source row / task	Difference in response content	Source label / prediction / score
70 / recommend foods worth buying	x recommends fresh truffles, emphasizing rarity and cooking, whereas y recommends aged cheese, emphasizing tasting and aging. The responses share a "gourmet experience" theme but recommend different concrete purchases.	0 / 1 / 0.999974
80 / explain in 30 words why one should use Vue.js	x adds reusable components and single-page applications, whereas y adds easy-to-learn syntax, a lightweight core, and an ecosystem. Their openings overlap strongly, but the later portions provide different reasons.	0 / 1 / 1.000000
39 / tell a joke	x uses wordplay in which removing the ears from "bear" leaves the letter B, whereas y uses a gummy-bear pun about a toothless bear. The source labels the pair equivalent, while the model predicts distinct.	1 / 0 / 0.193834
64 / suggest a four-character password	x gives the eight-character password `T35t#x4n`, whereas y gives the four-character password `T3#r`. The responses differ in whether they satisfy the stated length requirement, yet the source labels the pair equivalent.	1 / 0 / 0.000034

F Detailed Results for Relation Decisions, Use Response, and Set Retention

This appendix supplements the main text with use-specific relation decisions, full use-response results, family-level threshold transfer, and set-retention analyses. The controlled relation analyses reuse the 64 development cases, 128 test cases, and cached scores. The Purpose-Only Control, Canonicalization + Purpose Rule, and full use-response analyses are post-hoc interpretive analyses added after the primary results became available. The three-alternative extension and Natural-50 use their own data and statistical units. The implementation stage of each analysis is documented in Appendix B.

F.1 Use-Specific Relation Decisions and Full Use Response

Inputs and processing for the canonicalization baseline. The Canonicalization + Purpose Rule receives only the task and response strings q, x, y and operates independently of the data generator and constraint validator. It requires each response to contain exactly one JSON object with a unique `answer` key. Finite integers and decimals are mapped to canonical decimal numbers; sets are mapped to unordered numerical sets; paths preserve node order; and time intervals preserve start–end order. The baseline parses these objects directly from the visible text. It does not access structured task constraints or reference labels and does not determine whether an answer is feasible or optimal.

Using use-conditioned response pairs as the evaluation inputs, the Canonicalization + Purpose Rule successfully parses objects for all 2,176 test inputs. These inputs contain repeated response texts and therefore do not represent 2,176 independent responses. Of the 2,176 inputs, 2,048 have an explicit intended use and are handled according to the rules specified in the development text. The remaining 128 use-ambiguous inputs return 0, while their reference labels remain `null`, so they do not enter binary relation evaluation. Any parsing failure would likewise return 0 rather than being dropped; such a failure would therefore be treated as a prediction of distinctness and would affect the corresponding decision rates. Parsing records for the original scoring interfaces are unchanged: the two format failures for Qwen3-P continue to use their saved continuous scores.

Table F1: Relation decision rates under the two intended uses. Thresholds are those selected on the original development set. The Feasibility-Confirmation Use contains 768 valid equivalent relations. The Alternative-Preservation Use contains 256 equivalent and 512 distinct relations. All relations are drawn from the same 128 complete cases.

Scoring interface / control	Confirmation-use Equivalent Acceptance Rate	Preservation-use Equivalent Acceptance Rate	Preservation-use Distinct Rejection Rate
Jaccard	0.523	0.891	0.660
TF-IDF	0.165	0.434	0.969
STS	0.445	0.980	0.822
NoveltyBench	0.405	0.879	0.832
Qwen3-U	0.333	1.000	1.000
Qwen3-P	0.975	1.000	0.990
Canonicalization + Purpose Rule	1.000	1.000	1.000
Purpose-Only Control	1.000	0.000	1.000

Table F2: 95% case-level intervals for pooled-use decision rates and invalid-response rejection. Equivalent Acceptance Rate and Distinct Rejection Rate follow the definitions in the main text and should not be interpreted as precision and recall. Intervals with identical lower and upper endpoints arise from identical outcomes across the current fixed sample.

Scoring interface / control	Valid Equivalent Acceptance Rate [95% CI]	Valid Distinct Rejection Rate [95% CI]	Invalid-Response Rejection Rate [95% CI]
Jaccard	0.615 [0.602, 0.630]	0.660 [0.635, 0.684]	0.313 [0.277, 0.348]
TF-IDF	0.232 [0.208, 0.259]	0.969 [0.953, 0.982]	0.605 [0.563, 0.648]
STS	0.579 [0.556, 0.601]	0.822 [0.781, 0.865]	0.551 [0.496, 0.605]
NoveltyBench	0.523 [0.508, 0.539]	0.832 [0.811, 0.854]	0.461 [0.414, 0.508]
Qwen3-U	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]
Qwen3-P	0.981 [0.972, 0.990]	0.990 [0.982, 0.998]	0.699 [0.670, 0.729]
Canonicalization + Purpose Rule	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]	0.500 [0.500, 0.500]
Purpose-Only Control	0.750 [0.750, 0.750]	1.000 [1.000, 1.000]	0.500 [0.500, 0.500]

Use-specific and pooled-use decisions. Table F1 reports decision rates separately under the Feasibility-Confirmation Use and Alternative-Preservation Use. Among valid response pairs, the Feasibility-Confirmation Use contains only the equivalent class. Its standalone AUC and Distinct Rejection Rate are therefore undefined.

The Canonicalization + Purpose Rule correctly handles valid–valid relations under both intended uses, but its rejection rate in the separate Invalid-Response Diagnostic is 0.500. Under the Feasibility-Confirmation Use, the rule directly accepts without validating the task constraints. Under the Alternative-Preservation Use, an invalid response typically parses to a canonical object different from that of the valid response and is therefore rejected. This result separates canonical-object comparison from task-constraint validation: correctly determining relations among valid responses does not imply that the rule solves the underlying task or verifies response validity.

Full use response under fixed responses. As defined in Appendix A.1, each case contains four different-object response pairs whose reference relation should change with intended use and two same-object pairs whose reference relation should remain equivalent. A response pair is counted as correct in this analysis only if its judgments are correct under both uses. For each case, F and H are first averaged over the four changing-relation pairs and the two unchanged-relation pairs, respectively, and are then aggregated across cases.

Table F3: Full use response with 95% case-level intervals. F measures joint correctness for relations that should change with intended use, and H measures joint correctness for relations that should remain unchanged. Both statistics aggregate over the 128 complete cases.

Scoring interface / control	**Flip F [95% CI]**	**Hold H [95% CI]**
Jaccard	0.000 [0.000, 0.000]	0.891 [0.859, 0.918]
TF-IDF	0.000 [0.000, 0.000]	0.434 [0.383, 0.488]
STS	0.000 [0.000, 0.000]	0.980 [0.965, 0.996]
NoveltyBench	0.000 [0.000, 0.000]	0.879 [0.859, 0.898]
Qwen3-U	0.000 [0.000, 0.000]	1.000 [1.000, 1.000]
Qwen3-P	0.953 [0.932, 0.971]	1.000 [1.000, 1.000]
Canonicalization + Purpose Rule	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]
Purpose-Only Control	1.000 [1.000, 1.000]	0.000 [0.000, 0.000]

Reporting both quantities is important. Evaluating only the relations that should change would fail to detect a rule that simply changes every judgment whenever the intended use changes, including judgments that should remain invariant.

Table F3 shows that the Purpose-Only Control correctly handles every relation that should change with intended use but fails on every relation that should remain unchanged. Qwen3-P correctly handles 488 of the 512 response pairs whose relations should change and all response pairs whose relations should remain unchanged.

The intervals above use the same 1,000 family-stratified complete-case bootstrap samples, with models, prompts, parsing rules, and development thresholds held fixed. For AUC calculations, case/class weights are combined with case multiplicities from the bootstrap rather than treating individual response pairs as independent observations. The Canonicalization + Purpose Rule and Purpose-Only Control are included to interpret the information structure of the current controlled suite. Their degenerate intervals should not be interpreted as estimates of general error rates on natural text.

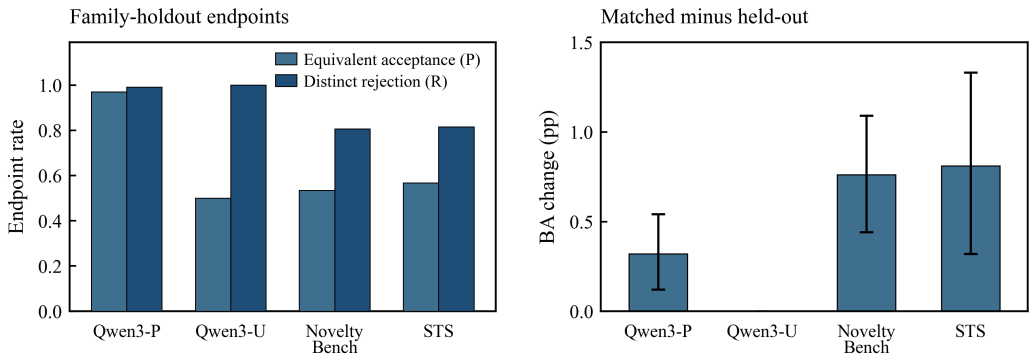
F.2 Family-Level Threshold Transfer with Equal Development Size

This analysis holds the scoring models and continuous scores fixed and refits only the decision thresholds. For each target task family, evaluation uses the original 32 test cases. The family-holdout branch uses 16 development cases from each of the other three task families, whereas the matched-development branch uses 12 development cases from each of all four families, including the target family. Both branches therefore use exactly 48 development cases. Here, *family holdout* means only that the target family is excluded from threshold development; it does not imply that the underlying model was never exposed to that task family during pretraining.

The matched-development subsets are selected by SHA-256 ordering using the seed `20260910`, the repetition index, and the anonymous case identifier. Target-test labels are not used in this selection. For each test case, results from the ten matched-development subsets are averaged before aggregation. These ten subsets therefore do not increase the number of independent test observations.

Both branches construct candidate thresholds from the unique development scores, the midpoints between adjacent scores, 0, 1, and `nextafter(1,`∞`)`. Candidates within 10^{-12} of the maximum BA are retained; among these, candidates within 10^{-12} of the best lower decision rate are retained; the lower median of the remaining sorted candidates is then selected. Threshold fitting first gives equal weight to task families and then equal weight to cases and classes within each family. Target-test scores are never used to construct candidate thresholds.

This pooled-use evaluation computes the Equivalent Acceptance Rate and Distinct Rejection Rate separately and defines Balanced Accuracy (BA) as their mean. Relative to the family-holdout branch, matched development increases the four-family macro-average BA of Qwen3-P by 0.32 percentage points (95% CI: 0.12–0.54 percentage points). The corresponding increases for NoveltyBench and



(a) **(b)**

Figure S2: Family holdout and equal-size matched development. The left panel reports point estimates of the Equivalent Acceptance Rate and Distinct Rejection Rate under family-holdout thresholds. The right panel reports paired BA changes for matched development relative to family holdout, with 95% intervals in percentage points (pp). The two branches use the same number of development cases and the same test objects; scoring models and continuous scores remain fixed, and only thresholds are re-estimated. The `Family-holdout` label in the figure refers to the family-holdout branch defined above.

Table F4: Equivalent Acceptance Rate / Distinct Rejection Rate by task family. Each cell reports the two decision-rate endpoints averaged over the original 32 test cases. For the matched-development branch, results are first averaged over the ten fixed development subsets. "Holdout" means only that the target family is excluded from threshold fitting; it does not imply absence of that family from model pretraining.

Scoring interface / development setting	Prime	Path	Scheduling	Subset sum
Qwen3-P / holdout	0.992/1.000	1.000/1.000	0.984/0.984	0.902/0.977
Qwen3-P / matched	0.991/0.998	1.000/1.000	0.980/0.986	0.936/0.974
Qwen3-U / holdout	0.500/1.000	0.500/1.000	0.500/1.000	0.500/1.000
Qwen3-U / matched	0.500/1.000	0.500/1.000	0.500/1.000	0.500/1.000
NoveltyBench / holdout	0.309/0.930	0.809/0.383	0.496/0.992	0.523/0.922
NoveltyBench / matched	0.347/0.905	0.761/0.479	0.501/0.988	0.532/0.912
STS / holdout	0.469/0.984	0.438/1.000	0.617/0.766	0.746/0.508
STS / matched	0.494/0.981	0.481/0.988	0.597/0.802	0.742/0.506

STS are approximately 0.76 and 0.81 percentage points, respectively. Qwen3-U is unchanged: in this pooled-use evaluation, its Equivalent Acceptance Rate remains 50% and its Distinct Rejection Rate remains 100% (Figure S2).

Under family holdout, Qwen3-P's lowest Equivalent Acceptance Rate is 0.902 and its lowest Distinct Rejection Rate is 0.977; both occur for the subset-sum family. Table F5 adds complete-case intervals for all task families so that macro-average changes and family-specific behavior can be examined separately.

Matched development affects the scoring interfaces differently. This comparison holds the total number of development cases, statistical weighting, and test objects constant, but it necessarily changes the task-family composition of the development data. The observed differences therefore cannot be further decomposed here into mechanism effects and template effects. The development cases, fitted thresholds, and predictions for each fold are retained in `transfer_folds.jsonl` and `transfer_predictions.jsonl`, allowing the decision rates for each target family to be recomputed.

Table F5: Family-level decision rates with 95% complete-case intervals. Each row is based on the 32 test cases for the corresponding target family. The matched-development branch keeps the ten development subsets fixed. The table adds intervals to the point estimates in Table F4 without fitting any additional models or thresholds.

Scoring interface	Task family	Development setting	Equivalent acceptance rate [95% CI]	Distinct rejection rate [95% CI]
Qwen3-P	Prime	holdout	0.992 [0.980, 1.000]	1.000 [1.000, 1.000]
Qwen3-P	Path	holdout	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]
Qwen3-P	Scheduling	holdout	0.984 [0.965, 1.000]	0.984 [0.961, 1.000]
Qwen3-P	Subset sum	holdout	0.902 [0.871, 0.934]	0.977 [0.945, 1.000]
Qwen3-P	Prime	matched	0.991 [0.979, 1.000]	0.998 [0.995, 1.000]
Qwen3-P	Path	matched	1.000 [1.000, 1.000]	1.000 [1.000, 1.000]
Qwen3-P	Scheduling	matched	0.980 [0.958, 0.996]	0.986 [0.963, 1.000]
Qwen3-P	Subset sum	matched	0.936 [0.904, 0.962]	0.974 [0.945, 0.998]
Qwen3-U	Prime	holdout	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Path	holdout	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Scheduling	holdout	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Subset sum	holdout	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Prime	matched	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Path	matched	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Scheduling	matched	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
Qwen3-U	Subset sum	matched	0.500 [0.500, 0.500]	1.000 [1.000, 1.000]
NoveltyBench	Prime	holdout	0.309 [0.277, 0.348]	0.930 [0.891, 0.969]
NoveltyBench	Path	holdout	0.809 [0.781, 0.844]	0.383 [0.313, 0.438]
NoveltyBench	Scheduling	holdout	0.496 [0.477, 0.512]	0.992 [0.977, 1.000]
NoveltyBench	Subset sum	holdout	0.523 [0.488, 0.559]	0.922 [0.875, 0.961]
NoveltyBench	Prime	matched	0.347 [0.308, 0.389]	0.905 [0.865, 0.948]
NoveltyBench	Path	matched	0.761 [0.733, 0.792]	0.479 [0.416, 0.534]
NoveltyBench	Scheduling	matched	0.501 [0.486, 0.512]	0.988 [0.971, 0.998]
NoveltyBench	Subset sum	matched	0.532 [0.503, 0.561]	0.912 [0.867, 0.952]
STS	Prime	holdout	0.469 [0.434, 0.500]	0.984 [0.961, 1.000]
STS	Path	holdout	0.438 [0.398, 0.469]	1.000 [1.000, 1.000]
STS	Scheduling	holdout	0.617 [0.570, 0.672]	0.766 [0.656, 0.859]
STS	Subset sum	holdout	0.746 [0.676, 0.813]	0.508 [0.375, 0.648]
STS	Prime	matched	0.494 [0.475, 0.509]	0.981 [0.958, 1.000]
STS	Path	matched	0.481 [0.459, 0.500]	0.988 [0.969, 1.000]
STS	Scheduling	matched	0.597 [0.558, 0.646]	0.802 [0.707, 0.881]
STS	Subset sum	matched	0.742 [0.676, 0.807]	0.506 [0.377, 0.638]

Table F6: Macro-average decision-rate changes for equal-size matched development relative to family holdout. Each difference is matched development minus family holdout and is reported on the proportion scale. Intervals use paired resampling of the same test cases. Positive values indicate a higher corresponding decision rate when development data include the target family.

Scoring interface	Equivalent Acceptance Rate difference [95% CI]	Distinct Rejection Rate difference [95% CI]	BA difference [95% CI]
Qwen3-P	0.0072 [0.0033, 0.0110]	-0.0008 [-0.0021, 0.0008]	0.0032 [0.0012, 0.0054]
Qwen3-U	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]
NoveltyBench	0.0009 [-0.0069, 0.0090]	0.0143 [0.0035, 0.0260]	0.0076 [0.0044, 0.0109]
STS	0.0112 [0.0021, 0.0216]	0.0049 [-0.0035, 0.0150]	0.0081 [0.0032, 0.0133]

Table F7: Thresholds selected under the 48-case development settings. The matched-development column reports the minimum-to-maximum range across the ten selected thresholds. These ranges describe variation across the fixed development subsets and are not confidence intervals. Evaluation uses the full-precision threshold values stored in JSON.

Fixed scoring interface	Target family	Family-holdout threshold	Matched-development threshold range
Qwen3-P	Prime	0.497087	0.476281–0.499996
Qwen3-P	Path	0.497087	0.476281–0.499996
Qwen3-P	Scheduling	0.497087	0.476281–0.499996
Qwen3-P	Subset sum	0.499996	0.476281–0.499996
Qwen3-U	Prime	0.727929	0.549207–0.749567
Qwen3-U	Path	0.749567	0.549207–0.749567
Qwen3-U	Scheduling	0.749567	0.549207–0.749567
Qwen3-U	Subset sum	0.749567	0.549207–0.749567
NoveltyBench	Prime	0.141768	0.126816–0.140481
NoveltyBench	Path	0.126920	0.126816–0.140481
NoveltyBench	Scheduling	0.134904	0.126816–0.140481
NoveltyBench	Subset sum	0.134904	0.126816–0.140481
STS	Prime	0.732902	0.718090–0.741842
STS	Path	0.741842	0.718090–0.741842
STS	Scheduling	0.717019	0.718090–0.741842
STS	Subset sum	0.727120	0.718090–0.741842

F.3 Development Selection Objectives and Coverage of Tied Schemes

This analysis changes only the development-selection objective while retaining the same six scoring interfaces, two linkage rules, and original development thresholds. The Intended-Use AUC Scheme selects the scoring interface according to Intended-Use AUC computed from valid equivalent and valid distinct relations under the Alternative-Preservation Use and then applies prespecified single linkage. Because AUC itself is independent of the linkage rule, Table F8 reports both linkage variants when displaying the complete set of development candidates and ties.

The Intended-Use Post-Grouping BA Scheme instead jointly evaluates the scoring interface and linkage rule using case-equal Post-Grouping BA computed from the co-cluster relation after grouping. Specifically, Post-Grouping BA is the mean of the post-grouping Equivalent Acceptance Rate and Distinct Rejection Rate. For both selection rules, any final tie is resolved by deterministic lexicographic ordering of the model and linkage names. Test results are not used in tie-breaking, and development Functional Coverage is not included in either the Intended-Use AUC or Intended-Use Post-Grouping BA selection objective.

Under the Intended-Use AUC development criterion, four interface/linkage combinations are tied at the maximum value: both linkage variants of TF-IDF and both linkage variants of Qwen3-U. Under the Intended-Use Post-Grouping BA criterion, five candidates are tied at the maximum: those four candidates plus Qwen3-P with complete linkage. The same five candidates are also tied under the primary diagnostic objective before its later tie-breaking stages are applied.

When evaluated under the original fixed response order, the test Functional Coverage of these development-tied candidates ranges from 0.8984 to 1.0000. Deterministic name-based sorting determines which candidate is ultimately selected when the development criterion remains tied, but a development-set tie does not imply equal test coverage.

In this setting, the intended-use-based schemes and the fixed complete-linkage rules attain the same Functional Coverage as the Diagnostic Scheme, yielding paired coverage differences of zero. Relative to the Pooled-AUC Scheme, the Diagnostic Scheme reduces the incorrect-merge rate by 0.0391, with

Table F8: Intended-use performance of the twelve development candidates. Pairwise BA is computed from pairwise decisions under the original development thresholds, whereas Post-Grouping BA is computed from the co-cluster relation after applying the specified linkage rule. Development Functional Coverage is reported descriptively and does not enter either the Intended-Use AUC or Intended-Use Post-Grouping BA selection objective.

Scoring interface / linkage	Intended-Use AUC	Pairwise BA	Post-Grouping BA	Development Functional Coverage
Jaccard / single	0.83739	0.746	0.584	0.859
Jaccard / complete	0.83739	0.746	0.783	0.930
TF-IDF / single	1.00000	1.000	1.000	1.000
TF-IDF / complete	1.00000	1.000	1.000	1.000
STS / single	0.93585	0.916	0.855	0.938
STS / complete	0.93585	0.916	0.900	0.953
NoveltyBench / single	0.94867	0.889	0.797	0.953
NoveltyBench / complete	0.94867	0.889	0.947	1.000
Qwen3-U / single	1.00000	1.000	1.000	1.000
Qwen3-U / complete	1.00000	1.000	1.000	1.000
Qwen3-P / single	0.99991	0.982	0.945	0.992
Qwen3-P / complete	0.99991	0.982	1.000	1.000

Table F9: Paired test differences between the Diagnostic Scheme and each comparator under the original fixed response order. The analysis contains 128 cases, with two responses retained per case. Each entry is a proportion difference defined as the Diagnostic Scheme minus the comparator. A negative incorrect-merge-rate difference therefore indicates fewer incorrect merges under the Diagnostic Scheme. Development-selected schemes and thresholds are held fixed when the intervals are computed. Because this table uses the original fixed response order, its coverage values differ from the all-order averages reported in Table 2.

Comparator	Incorrect-merge-rate difference [95% CI]	Incorrect-split-rate difference [95% CI]	Functional Coverage difference [95% CI]
Pooled-AUC Scheme	-0.0391 [-0.0705, -0.0078]	0.0000 [0.0000, 0.0000]	0.0039 [0.0000, 0.0117]
Intended-Use AUC Scheme	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]
Intended-Use Post-Grouping BA Scheme	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]
Fixed Complete-Linkage Scheme	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]
Qwen3-U / single linkage	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]
Qwen3-U / complete linkage	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]	0.0000 [0.0000, 0.0000]

a 95% interval of [-0.0705, -0.0078]. Its corresponding Functional Coverage difference is 0.0039, with a 95% interval of [0.0000, 0.0117].

These results illustrate that reducing local grouping errors and increasing set-level coverage are not equivalent objectives. A scheme can reduce incorrect merges without producing a clearly separated coverage difference under the current tasks, response order, and selection budget. Local relation and grouping errors should therefore be reported separately from their consequences for Functional Coverage.

Table F10: Fixed-order and order-averaged Functional Coverage with 95% case-level intervals. The analysis contains 128 independent test tasks, and coverage is reported on the 0–1 scale. Each column uses the corresponding set of 1,000 family-stratified complete-task bootstrap samples, with the original development thresholds and continuous scores held fixed. The deterministic tie-breaking rules used during grouping and selection are described in Appendix F.5.

Scoring interface / control	**Original fixed order [95% CI]**	**Order-averaged [95% CI]**
Jaccard	0.9375 [0.9063, 0.9648]	0.9300 [0.9131, 0.9453]
TF-IDF	0.8984 [0.8633, 0.9336]	0.9030 [0.8936, 0.9128]
STS	0.9609 [0.9336, 0.9805]	0.9701 [0.9596, 0.9805]
NoveltyBench classifier	0.9844 [0.9687, 0.9961]	0.9785 [0.9740, 0.9824]
Qwen3-U	1.0000 [1.0000, 1.0000]	1.0000 [1.0000, 1.0000]
Qwen3-P	1.0000 [1.0000, 1.0000]	1.0000 [1.0000, 1.0000]
Purpose-Only Control	0.8477 [0.8047, 0.8867]	0.8333 [0.8333, 0.8333]
Canonicalization + Purpose Rule	1.0000 [1.0000, 1.0000]	1.0000 [1.0000, 1.0000]

F.4 Fixed-Order and Order-Averaged Coverage Under Complete Linkage

Table 1 in the main text fixes complete linkage, the original development thresholds, and a selection budget of $K = 2$, and reports order-averaged Functional Coverage for the six scoring interfaces and two interpretive controls. Table F10 places this quantity alongside the corresponding result under the original fixed response order.

The original fixed-order column is taken from the existing complete-linkage test records and preserves grouping and representative selection under the original response identifiers and positions. The order-averaged column instead reruns grouping and representative selection after each possible permutation of the four responses, using their new positions, and then averages Functional Coverage over all 24 permutations within each task.

Neither analysis rescored responses, refitted thresholds, or changed any operating rule in response to test results. The two columns therefore differ only in how input order—and the downstream grouping and representative-selection process that depends on that order—is handled. They do not differ in the underlying tasks, scoring models, continuous scores, development thresholds, linkage rule, or selection budget.

The Purpose-Only Control has higher Pooled AUC than Qwen3-U but lower Intended-Use AUC and lower Functional Coverage. The Canonicalization + Purpose Rule attains the same coverage as both Qwen3 interfaces under complete linkage. These comparisons further illustrate that pairwise ranking performance and final set utility need not vary together.

For the Purpose-Only Control, Functional Coverage is 0.8477 under the original fixed response order and 0.8333 after averaging over all 24 input orders. These values summarize different order conditions and should not be treated as interchangeable estimates of the same quantity. More generally, the differences between the two columns show that deterministic representative selection can make final coverage depend on response order even when scoring interfaces, thresholds, and linkage rules are held fixed.

The recomputation workflow for the original fixed-order analysis is `purpose-information`, with results stored in `output/current_information_controls/`. The workflow for the order-averaged analysis is `order-averaged-retention`, with results stored in `output/current_order_retention/`. The common invocation procedure is described in Appendix G.

F.5 Input Order, the Three-Alternative Extension, and Per-Task Loss

Aggregation and selection after permutation. For each response permutation, the original response identifiers are first mapped to their new positions, after which grouping and representative selection

Table F11: Task distribution and mean Avoidable Coverage Loss under the main loss conditions. Loss is defined as attainable coverage under the same budget minus actual order-averaged Functional Coverage and is reported in percentage points. Intervals resample independent tasks. "Mean loss" is averaged over all tasks in the corresponding condition, rather than only over tasks with nonzero loss. The balanced and imbalanced three-alternative conditions use the same 32 underlying tasks and are therefore not pooled as 64 independent observations.

Set condition / budget / scheme	Affected tasks	Task-family distribution among affected tasks	Mean loss [95% CI]
Original four-response set, $K = 2$ / Pooled-AUC Scheme	5/128	subset sum 3, scheduling 2	0.65 [0.13, 1.18]
Balanced three-alternative set, $K = 3$ / Pooled-AUC Scheme	5/32	subset sum 3, prime 1, scheduling 1	2.57 [0.63, 4.65]
Balanced three-alternative set, $K = 3$ / Intended-Use AUC Scheme	2/32	subset sum 2	0.97 [0.00, 2.43]
Imbalanced three-alternative set, $K = 3$ / Pooled-AUC Scheme	3/32	subset sum 1, prime 1, scheduling 1	1.88 [0.00, 3.75]

are rerun using those positions. Candidate merges are tie-broken sequentially by cross-cluster similarity, the earliest position represented by the two clusters, and a deterministic index. Single linkage uses the maximum cross-cluster similarity, whereas complete linkage uses the minimum. In both cases, a merge is accepted when the corresponding score is no lower than the fixed threshold. Results under the original fixed response order preserve the original implementation and are reported separately rather than being overwritten by the permutation analysis.

For every permutation, we store the resulting partition, retained response identifiers, incorrect merges, incorrect splits, and Functional Coverage. The 24 permutations in the original four-response setting, or the 720 permutations in the six-response three-alternative setting, are first summarized within each task. Aggregate statistics are then computed at the task level so that permutations of the same task are not treated as independent observations.

Three-alternative sets. The extension constructs three verified distinct canonical objects within each task: three prime numbers, three subsets satisfying the sum constraint, three equal-cost shortest paths, or three feasible time intervals. Each of the four task families contains eight tasks. Every task is instantiated under both a balanced `2/2/2` repetition pattern and an imbalanced `4/1/1` repetition pattern.

The generation rules, candidate responses, selection budgets, threshold-transfer procedure, and reported statistics for this extension were fixed before evaluator models were run on the new test sets. Thresholds and operating schemes are transferred from the established development-selection procedure and are not selected using the new test results.

The frozen extension protocol is recorded in `output/v79_set_extension_v1/COMMITMENT.json`. Input and output hashes for the retained analysis are recorded in `output/current_three_alternative_retention/COMPLETION.json`. Score files and their receipts are stored under `output/v79_set_extension_v1/scores/`. The release-wide file inventory and SHA-256 checksums are listed in `ARTIFACT_MANIFEST.json`. Structured summaries are stored in `output/current_three_alternative_retention/`.

The workflow for per-task loss tracing is `task-level-loss-tracing`. Complete verification records are stored in `output/current_loss_traces/`, containing 1,047,552 entries. The scheduling, balanced subset-sum, and prime-number examples reported in Table 4 correspond to `281c172b693c696c`, `bc650e03c3afe440-balanced`, and `0026d3fe4b31cfd5`, respectively. These identifiers allow the full task, canonical objects, thresholds, scores, grouping outcomes, and selections underlying each example to be checked.

The explanatory cases in Table 4 were selected after the aggregate results became available using the saved maximum-loss rule. They are used to illustrate how local relation decisions, grouping,

repetition patterns, input order, and the selection budget can combine to produce coverage loss. They are not used to estimate the prevalence of these mechanisms across tasks.

F.6 NATURAL-50 HUMAN REFERENCE AND COMPLETE-SET RESULTS

Human annotation and adjudication. Natural-50 contains 50 queries, each with six complete responses. Two non-author annotators with relevant research experience independently annotated Intended Use, response validity, functional units, and boundary categories. The blinded annotation package contained neither model outputs nor information about each query's role in threshold development, scheme selection, or final testing.

Twenty disputed queries were adjudicated by an author before any final-test evaluator results were viewed. Eight underwent construct re-review, and the remaining twelve underwent a `KEEP/REVISE` consistency review. The final human reference was then frozen, while both original annotation layers were retained in full.

Before adjudication, validity agreement was 289/300 (96.33%; Cohen's $\kappa = 0.6486$). Within the simple comparison domain, relation agreement was 466/585 (79.66%; $\kappa = 0.4244$). The macro-average Adjusted Rand Index (ARI) across the 49 queries for which it was computable was 0.7959, and 32 of the 40 comparable simple partitions matched exactly. The final reference passed completeness checks covering all 50 queries, 300 responses, and 750 explicitly annotated response pairs. No transitive closure was applied, and no model-result fields were included in the human reference.

Final test set. The 30 final-test queries contain 163 valid responses, 16 invalid responses, and 1 response with uncertain validity. Thirteen valid responses contain multiple functional units: 7 are annotated as multi-alternative and 6 as partially overlapping, distributed across 5 queries. Partial-overlap relations occur in 2 queries.

The number of functional units per query is distributed as follows: 10 queries contain 1 unit; 1 query contains 2 units; 1 contains 3 units; 2 contain 4 units; 1 contains 5 units; 12 contain 6 units; and 1 query each contains 16, 19, and 23 units.

When three responses are retained per query, the mean attainable coverage under the same budget is 77.64%. For 12 of the 30 queries, Functional Coverage depends on which three responses are selected. The frozen main analysis uses a selection budget of $K = 3$. Results for budgets $K = 2$ and $K = 4$ are post-hoc sensitivity analyses computed from the same frozen scores after the main results became available and are reported in Appendix H.2.

Pairwise evaluation. Table F12 evaluates response pairs from the final test set for which the human reference provides an unambiguous binary relation. Among the 360 eligible pairs, 106 are equivalent and 254 are distinct. An additional 18 partially overlapping pairs, 69 pairs involving an invalid response, and 3 uncertain relations are excluded from binary AUC and BA evaluation.

The 360 binary-labeled pairs come from 29 queries. Q044 contains no pair with a usable binary relation for this analysis but remains part of the complete-set Functional Coverage evaluation, which uses all 30 final-test queries.

Set evaluation at budget three. Natural-50 uses separate data for threshold development and scheme selection. Ten queries are first used to determine one threshold for each scoring interface. A separate set of nine eligible queries is then used for scheme selection.

The Natural-50 Intended-Use AUC Scheme selects the scoring interface by pairwise Intended-Use AUC and uses fixed single linkage. The Natural-50 Diagnostic Scheme instead compares scoring-interface/linkage combinations. It first maximizes the lower of the post-grouping Equivalent Acceptance Rate and Distinct Rejection Rate, and then resolves remaining ties using development Functional Coverage, Post-Grouping BA, and deterministic name ordering. The task-family stratification used in the controlled experiments is not applied to Natural-50.

Table F13 compares five frozen schemes. Functional Coverage is computed from the union of functional units contained in the selected valid responses. Attainable coverage under the same budget is obtained by enumerating every three-response subset of the same six candidates. Avoidable Coverage Loss is the difference between this same-budget attainable coverage and the actual Functional

Table F12: Natural-50 pairwise relations and performance under natural-development thresholds. Intervals are obtained by resampling queries rather than individual response pairs. P/R denotes Equivalent Acceptance Rate / Distinct Rejection Rate. Thresholds are determined from the natural development data. The table displays thresholds at manuscript precision, whereas all computations use the saved full-precision values.

Scoring interface	AUC [95% CI]	BA [95% CI]	P/R	Natural development threshold
Jaccard	0.752 [0.614, 0.865]	0.642 [0.541, 0.738]	0.885/0.398	0.167275
TF-IDF	0.683 [0.529, 0.828]	0.625 [0.515, 0.732]	0.469/0.781	0.927230
STS	0.745 [0.618, 0.870]	0.717 [0.612, 0.811]	0.918/0.516	0.587194
NoveltyBench	0.780 [0.652, 0.892]	0.723 [0.630, 0.811]	0.699/0.746	0.146225
Qwen3-U	0.797 [0.686, 0.897]	0.700 [0.595, 0.796]	0.812/0.588	0.935086
Qwen3-P	0.685 [0.551, 0.809]	0.654 [0.541, 0.763]	0.744/0.564	0.999551

Table F13: Query-level Natural-50 results for frozen schemes at budget three. All five schemes use the same final-test queries, the same selection budget, and the same frozen response order. The Controlled Pooled-AUC Transfer Scheme and Fixed Complete-Linkage Scheme retain controlled-development thresholds, whereas the Natural-50 Intended-Use AUC Scheme and Natural-50 Diagnostic Scheme use natural-development thresholds. The First-K Baseline directly retains the first three responses. Functional Coverage is reported as a percentage, and Avoidable Coverage Loss relative to the same-budget optimum is reported in percentage points. Intervals are based on 1,000 query-level bootstrap resamples; between-scheme differences use paired query resamples.

Scheme	Scoring interface / linkage	Functional Coverage (%) [95% CI]	Avoidable Coverage Loss (pp) [95% CI]	Queries with loss
Controlled Pooled-AUC Transfer Scheme	Qwen3-P / single linkage	69.28 [59.21, 78.74]	8.36 [1.94, 16.03]	6/30
Natural-50 Intended-Use AUC Scheme	Qwen3-P / single linkage	73.00 [64.59, 80.97]	4.64 [1.11, 9.37]	5/30
Natural-50 Diagnostic Scheme	Qwen3-P / complete linkage	71.29 [62.46, 80.13]	6.35 [1.71, 11.83]	6/30
Fixed Complete-Linkage Scheme	Qwen3-P / complete linkage	69.28 [59.21, 78.74]	8.36 [1.94, 16.03]	6/30
First-K Baseline	none / input order	69.28 [59.21, 78.74]	8.36 [1.94, 16.03]	6/30

Coverage, and therefore measures omissions that could have been avoided without changing either the candidate responses or the selection budget.

When the six scoring interfaces are evaluated with their natural-development thresholds and single linkage, budget-three Functional Coverage ranges from 69.28% to 73.00%, with the observed ordering Qwen3-P, Qwen3-U, TF-IDF, Jaccard, NoveltyBench, and STS. This ordering differs from the pairwise AUC ordering in Table F12. The comparison involves six fixed interfaces under the current evaluation setting and is not interpreted as a stable overall ranking across methods.

The test-set pairwise ranking in Table F12 and the development-based scheme selection underlying Table F13 are also distinct results. In particular, the interface used by the Natural-50 Intended-Use AUC Scheme is selected from development data and is not reselected according to the final-test AUC values in Table F12.

In Table F13, the Natural-50 Intended-Use AUC Scheme exceeds the First-K Baseline in Functional Coverage by 3.72 percentage points, with a 95% paired interval of [0, 8.18]. Its coverage difference relative to the Natural-50 Diagnostic Scheme is 1.71 percentage points, with an interval of [0, 4.25].

The Natural-50 Diagnostic Scheme exceeds the First-K Baseline by 2.01 percentage points, with an interval of [0, 5.32]. All three intervals touch zero.

The 8.36-percentage-point shortfall for the Controlled Pooled-AUC Transfer Scheme represents a different quantity. It is the scheme's Avoidable Coverage Loss relative to the optimum attainable from the same candidates under the same budget, not a performance improvement relative to the First-K Baseline or another selection scheme.

Per-query loss and implementation checks. The original loss records are stored in `output/current_natural50_final/NATURAL50_LOSS_TRACES.jsonl`. A post-opening audit corrected explanatory mechanism labels; the corrected records are stored in `output/current_natural50_final/postopen_correction_001/NATURAL50_LOSS_TRACES_CORRECTED.jsonl`. The correction did not change the numerical estimands or scheme selections. A separate Qwen batch reproducibility check did not change the reported threshold decisions, partitions, representative selections, or numerical results.

Across 29 scheme–query records in the corrected traces, representative truncation occurs 26 times, cross-unit incorrect acceptance 19 times, single-linkage propagation 3 times, and invalid-response selection 5 times. Incorrect splitting and aggregation fragmentation each occur once. Multiple mechanisms may occur in one record, and repeated appearances of a query under different schemes are not independent observations. These counts are post-hoc explanatory labels; the frozen numerical results remain unchanged.

G Research Materials and Result Recomputation

The accompanying code artifact is a cached-score analysis artifact. It preserves controlled-task instances and reference relations, Natural-50 source responses and human annotation layers, evaluator continuous scores, development-selection records, and query-level results. From these saved inputs it recomputes threshold fitting, relation decisions, response grouping, representative selection, Functional Coverage, Avoidable Coverage Loss, and the paper tables. It does not download model weights or call an evaluator model. Recreating the continuous scores requires the separately licensed model weights and inference environments described in Appendix C.2 and is outside the scope of the artifact.

The tested environment is CPython 3.13 on Windows AMD64. Exact versions of the direct Python dependencies are pinned in `requirements.txt`; broader package constraints remain in `pyproject.toml`. The artifact contains a SHA-256 inventory, `ARTIFACT_MANIFEST.json`, and a one-command verifier. After installing the dependencies, the complete packaged verification is run with

```
python verify_release.py
```

The verifier checks the artifact inventory, runs the packaged scientific tests, executes all eleven registered workflows in fresh output directories, and checks deterministic outputs and headline manuscript values. The workflows cover controlled relations, purpose-information controls, 24-order retention, the 32-task three-alternative extension with 720 orders, task-level loss tracing, development-selection robustness, external natural-pair analysis, the frozen Natural-50 analysis, the post-hoc Natural-50 analyses in Appendix H, and final table generation. They can be listed with

```
python main.py list
```

For example, the Natural-50 development selection, query-level Functional Coverage, and loss records can be recomputed from the saved scores with

```
python main.py reproduce --workflow natural50-final
--output-dir
tmp/reproductions/natural50_check_01
```

The output directory must be a new project-relative path under `tmp/reproductions/`. Recomputed outputs, input checksums, and execution status are written there without overwriting the frozen inputs or authoritative results. Model and interface identifiers, truncation settings, prompts,

numerical readout settings, and score checksums are retained in the configurations, score manifests, and per-interface receipts; the top-level artifact manifest records the hash of every released file.

For Natural-50, the selected source responses are stored in `data/current_natural50/source_responses.csv`; blinded mappings, original A/B annotation layers, agreement records, the final adjudicated reference, and its validation records are stored under `annotation/natural50/`. Development and test scores are physically separated under `output/current_natural50_scores_v2/`, and the main query-level results are stored under `output/current_natural50_final/`. The public source commit and the data-redistribution boundary are documented in `DATA_AND_LICENCE_NOTES.md`.

The post-hoc Natural-50 analyses are available as two further offline workflows. Appendix H.1–H.3 can be recomputed with `natural50-posthoc-sensitivity`; Appendix H.4 uses `natural50-posthoc-subsets`. Both use the released saved scores, original and adjudicated human references, and frozen development choices. Their structured reference results are under `output/current_natural50_posthoc_sensitivity/` and `output/current_natural50_posthoc_subsets/`. The historical subset plan is preserved byte-for-byte under `configs/NATURAL50_SIMPLE_SUBSET_PLAN.json`; the `frozen_inputs/` prefix in that plan points to files now stored at the artifact root with matching SHA-256 hashes. The README gives fresh-output commands for both workflows.

H Natural-50 Post-Hoc Sensitivity Analyses

This appendix examines the sensitivity of the Natural-50 results to the human reference, response order, selection budget, and selection rule, and further compares the two Qwen interfaces across subsets with different response structures. All analyses in this appendix were proposed after the main results became available. The packaged workflows use only frozen evaluator scores and existing human annotations; they do not invoke evaluator models again, refit thresholds, modify the original annotations, or alter the main-analysis scheme. The results obtained with budget three and the frozen blinded response order remain the primary Natural-50 analysis.

H.1 Comparison Across Human References

To examine sensitivity to the human reference, we hold the scoring and selection settings fixed and compare the adjudicated Final reference with the two original annotation layers. The original A/B annotations encode grouping assignments and boundary markers rather than directly providing the functional-unit sets required for coverage computation. They are therefore converted into functional-unit sets using predefined mechanical rules before coverage is evaluated.

The conversion uses only information explicitly present in the original annotations. Simple grouping codes are mapped to query-local functional units. For responses marked `MULTI`, multiple functional units are created only when the existing literal-list parser identifies at least two visible items. Original annotation B marks Q044 as `PARTIAL_OVERLAP` without an explicit functional-unit set; consequently, Q044 is excluded only from set-level coverage calculations that use the B-derived reference. The conversion introduces no new semantic relations and does not relabel the original annotations.

Across all three references, Qwen3-U has the highest pairwise AUC, whereas Qwen3-P has the highest Functional Coverage under the corresponding fixed-threshold single-linkage evaluation. The same relationship is observed when the comparison is restricted to the 29 queries shared by all three references. Differences in validity judgments, grouping assignments, and `MULTI` annotations nevertheless change the absolute coverage values across references.

We additionally partition the 30 final-test queries according to whether the annotators disagreed before adjudication. Eleven queries contain at least one pre-adjudication disagreement, including six with selection space under budget three; the remaining 19 queries are undisputed. Among the undisputed queries, Qwen3-P has both the highest pairwise AUC and the highest Functional Coverage. Among the disputed queries, NoveltyBench has the highest pairwise AUC, while Qwen3-P and Qwen3-U attain the same Functional Coverage.

The full-sample reversal between the interface with the highest pairwise AUC and the interface with the highest Functional Coverage therefore emerges after the disputed and undisputed query groups

Table H1: Functional Coverage under alternative human references. In this table and Figure S3, the "target scheme" and "diagnostic scheme" refer to the Natural-50 Intended-Use AUC Scheme and Natural-50 Diagnostic Scheme, respectively. All schemes use the frozen response order and a selection budget of $K = 3$. Final is the primary adjudicated reference; Original A and Original B are mechanically converted according to the predefined rules above and are used only for sensitivity analysis. Final and Original A include all 30 final-test queries, whereas Original B includes 29 because Q044 lacks an explicit functional-unit set. Attainable coverage and Functional Coverage are reported as percentages. The final column reports the paired query-level coverage difference between the Natural-50 Intended-Use AUC Scheme and the First-K Baseline in percentage points; bracketed values are 95% paired bootstrap intervals.

Reference	Queries	Attainable coverage (%)	Controlled transfer (%)	Target scheme (%)	Diagnostic scheme (%)	Target − First-K (pp) [95% CI]
Final	30	77.64	69.28	73.00	71.29	3.72 [0, 8.18]
Original A	30	76.91	69.06	71.94	71.06	3.55 [0, 8.04]
Original B	29	66.64	57.93	60.92	60.01	3.67 [0, 8.03]

are combined rather than appearing in the same form within both groups. More broadly, the complete pairwise-AUC and Functional Coverage orderings remain different within each group. The paired coverage difference between the Natural-50 Intended-Use AUC Scheme and the First-K Baseline is 2.27 percentage points [0, 6.82] on disputed queries and 4.55 percentage points [0, 11.60] on undisputed queries.

H.2 Response Order and Selection Budget

To examine sensitivity to response order, we enumerate all 720 permutations of the six responses for every final-test query. For each permutation, clustering and representative selection are rerun using the permuted positions. Functional Coverage is first averaged over all 720 response orders within each query and is then aggregated across queries.

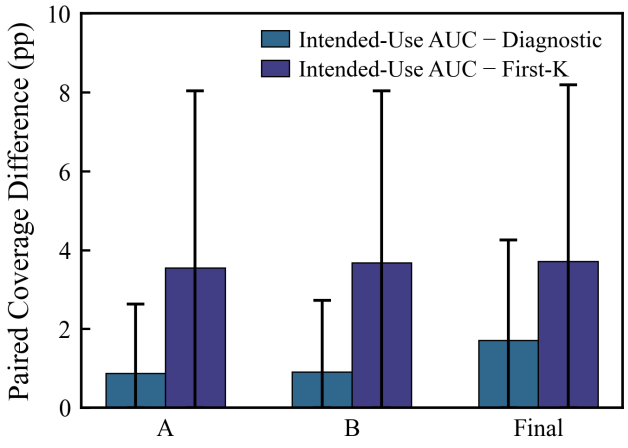
Under a selection budget of $K = 3$ with single linkage, the order-averaged Functional Coverage values for the six scoring interfaces are 72.15% for Qwen3-P, 71.07% for Jaccard, 71.04% for STS, 71.00% for Qwen3-U, 70.14% for TF-IDF, and 69.10% for NoveltyBench. This ordering differs from the pairwise-AUC ordering, showing that the discrepancy between pairwise ranking performance and set-level coverage is not confined to the original frozen blinded response order. For every scoring interface, Functional Coverage changes with input order on at least 10 of the 30 final-test queries.

We next vary the selection budget while holding the saved scores, natural-development thresholds, and single-linkage rule fixed. At $K = 2$, STS has the highest order-averaged Functional Coverage, at 58.77%. At $K = 3$ and $K = 4$, Qwen3-P has the highest order-averaged coverage, at 72.15% and 82.10%, respectively. Mean attainable coverage under the same budget is 67.55% for $K = 2$, 77.64% for $K = 3$, and 86.61% for $K = 4$.

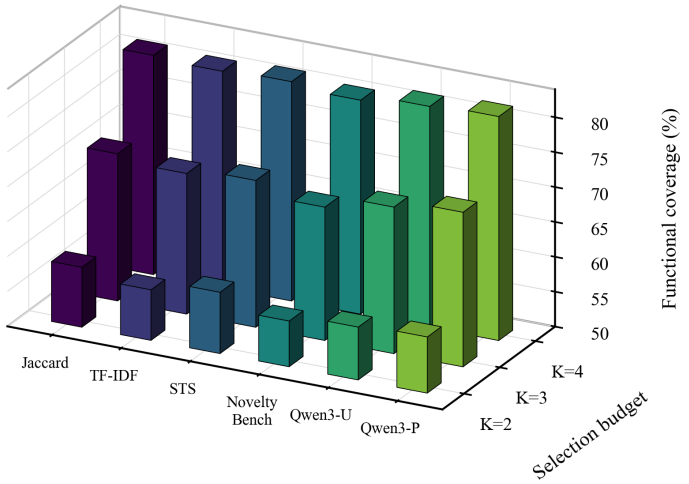
These budget comparisons are post-hoc sensitivity analyses and are not used to reselect the main-analysis budget. The primary Natural-50 analysis remains the frozen blinded response order with $K = 3$. More generally, the change in coverage ordering across budgets shows that Selection Compatibility depends not only on pairwise scores and grouping decisions but also on the number of responses that the downstream selector is allowed to retain.

H.3 Non-Clustering Selection from Continuous Scores

As an additional sensitivity analysis, we apply a `farthest-first` selection rule that bypasses thresholding and clustering and allocates the selection budget directly from the continuous equivalence scores. The rule was fixed before this post-hoc analysis was run. It first selects the earliest response in the current input order. At each subsequent step, for every unselected response, it computes the maximum equivalence score between that response and any response already selected. The response with the smallest such maximum is selected next, with ties resolved by input order.



(a) Human-reference sensitivity



(b) Coverage by selection budget

Figure S3: Natural-50 sensitivity to the human reference and selection budget. The upper panel compares the Natural-50 Intended-Use AUC Scheme with the First-K Baseline and the Natural-50 Diagnostic Scheme under the adjudicated Final reference and the mechanical projections of Original A and Original B. Values are paired query-level coverage differences, and all 95% intervals include zero. "Paired Coverage Difference" denotes a query-paired coverage-rate difference rather than a response-pair metric. The lower panel holds continuous scores, thresholds, and single linkage fixed and reports order-averaged Functional Coverage for the six scoring interfaces at budgets $K = 2$, $K = 3$, and $K = 4$. Mean attainable coverage for each budget is reported in Appendix H.2.

Under the frozen blinded response order and a budget of $K = 3$, Qwen3-U attains the highest Functional Coverage at 73.83%. After averaging over all response orders, however, NoveltyBench attains the highest coverage at 72.89%, followed by Qwen3-U at 71.76% and Qwen3-P at 71.74%.

Changing the selection rule therefore changes the ordering of the scoring interfaces by Functional Coverage. In particular, the Qwen3-U/Qwen3-P ordering observed under the threshold-and-clustering pipeline does not persist under every selection rule. At the same time, replacing thresholded clustering with direct continuous-score selection does not make the pairwise-AUC and set-level coverage orderings universally agree. These results further indicate that the mapping from Pairwise Evidence to Set Utility depends on the downstream selection procedure rather than on the pairwise scorer alone.

Table H2: Paired comparison on post-hoc comparable subsets. "Selection space" counts queries for which distinct three-response subsets can yield different Functional Coverage under $K = 3$. ΔAUC is Qwen3-U minus Qwen3-P on the original AUC scale; ΔFC is Qwen3-P minus Qwen3-U in percentage points. Brackets give paired query-level 95% bootstrap intervals. These descriptive post-hoc subsets are not independent validation sets. The final column lists the interfaces achieving the highest coverage within each subset.

Subset	N	**Space**	**Attainable (%)**	**ΔAUC [95% CI]**	**ΔFC (pp) [95% CI]**	**Highest FC**
All test	30	12	77.64	0.112 [0.025, 0.209]	1.30 [0, 3.91]	Qwen3-P
Simple	24	6	75.42	0.127 [0.028, 0.250]	0 [0, 0]	Qwen3-P = Qwen3-U = TF-IDF
Complex	6	6	86.51	0.150 [-0.194, 0.435]	6.52 [0, 19.57]	Qwen3-P
Simple-All-Valid	17	3	69.12	0.057 [-0.039, 0.182]	0 [0, 0]	Qwen3-P = Qwen3-U

The analyses in Appendices H.1–H.3 were computed from frozen evaluator scores and human annotations without invoking evaluator models, refitting thresholds, or modifying the reference. They remain post-hoc sensitivity analyses. The accompanying `natural50-posthoc-sensitivity` workflow reproduces their structured results; Appendix G specifies the packaged inputs and fresh-output procedure.

H.4 Paired Comparison of Qwen Interfaces and Post-Hoc Comparable Subsets

Direct paired comparison. This analysis fixes the Final human reference, natural-development thresholds, frozen blinded response order, single linkage, and a selection budget of $K = 3$, and directly compares Qwen3-U with Qwen3-P. We use 1,000 query-level paired bootstrap resamples with seed `20260910`. Within each bootstrap repeat, the same sampled queries are used for both interfaces and for both the pairwise-AUC and Functional Coverage comparisons.

Across the full 30-query final test set, 29 queries contribute usable binary relations to the pairwise-AUC calculation, while all 30 queries contribute to the Functional Coverage calculation. Qwen3-U exceeds Qwen3-P in pairwise AUC by 0.112, with a 95% paired interval of [0.025, 0.209]. In the opposite direction, the Functional Coverage difference, defined as Qwen3-P minus Qwen3-U, is 1.30 percentage points, with a 95% paired interval of [0, 3.91]. The entire observed coverage difference arises from a single query. Thus, the paired AUC interval supports higher pairwise AUC for Qwen3-U under the current evaluation setting, whereas the coverage-difference interval touches zero and does not establish a stable coverage advantage for Qwen3-P. This post-hoc comparison is descriptive and is not adjusted for multiple comparisons.

Definition of comparable subsets. To examine how response-set structure relates to the differences above, we additionally define post-hoc subsets of the Natural-50 final test set. The subset rule was written to `NATURAL50_SIMPLE_SUBSET_PLAN.json` before the subset-specific results were viewed, although the decision to conduct this subset analysis was itself made after the main results became available.

`Simple` requires resolved validity for every response, exactly one functional unit for each valid response, no `MULTI`, `PARTIAL`, or unknown-validity cases, and an unambiguous binary relation label for every valid–valid response pair. A `Simple` query may still contain invalid responses.

`Simple-All-Valid` imposes all `Simple` requirements and additionally requires all six responses to be valid. Consequently, every candidate entering set selection for this subset also lies within the binary pairwise-evaluation domain.

`Complex` contains all final-test queries that do not satisfy the `Simple` criteria.

Within `Simple`, the six interfaces have the following pairwise-AUC ordering: Qwen3-U, Novelty-Bench, STS, Jaccard, Qwen3-P, and TF-IDF. By contrast, Qwen3-U, Qwen3-P, and TF-IDF attain

Table H3: Performance of the six frozen scoring interfaces on the post-hoc subsets. Functional Coverage uses each interface's natural-development threshold, single linkage, the frozen blinded response order, and a selection budget of $K = 3$. Pairwise AUC is reported on the 0–1 scale, and Functional Coverage is reported as a percentage.

Subset	Interface	Pairwise AUC [95% CI]	Functional Coverage (%) [95% CI]
Simple	Jaccard	0.741 [0.595, 0.875]	71.46 [61.04, 82.50]
Simple	TF-IDF	0.613 [0.442, 0.779]	73.33 [62.92, 83.75]
Simple	STS	0.755 [0.601, 0.893]	72.29 [61.46, 83.33]
Simple	NoveltyBench	0.788 [0.674, 0.890]	72.29 [61.46, 83.33]
Simple	Qwen3-U	0.809 [0.689, 0.917]	73.33 [63.32, 83.75]
Simple	Qwen3-P	0.683 [0.538, 0.809]	73.33 [63.32, 83.75]
Complex	Jaccard	0.804 [0.721, 0.884]	65.13 [41.94, 86.15]
Complex	TF-IDF	1.000 [1.000, 1.000]	60.01 [31.47, 88.54]
Complex	STS	0.639 [0.356, 0.917]	57.24 [31.37, 82.84]
Complex	NoveltyBench	0.574 [0.433, 0.810]	57.97 [35.49, 79.96]
Complex	Qwen3-U	0.511 [0.267, 0.744]	65.13 [41.94, 86.15]
Complex	Qwen3-P	0.361 [0.209, 0.494]	71.65 [58.47, 87.96]
Simple-All-Valid	Jaccard	0.776 [0.625, 0.903]	67.65 [55.88, 79.41]
Simple-All-Valid	TF-IDF	0.666 [0.499, 0.843]	67.65 [55.88, 79.41]
Simple-All-Valid	STS	0.736 [0.554, 0.888]	67.65 [55.88, 79.41]
Simple-All-Valid	NoveltyBench	0.748 [0.580, 0.898]	67.65 [55.88, 79.41]
Simple-All-Valid	Qwen3-U	0.829 [0.713, 0.935]	69.12 [57.35, 80.88]
Simple-All-Valid	Qwen3-P	0.772 [0.648, 0.889]	69.12 [57.35, 80.88]

the same highest Functional Coverage. The Functional Coverage difference between Qwen3-P and Qwen3-U is exactly zero in both `Simple` and `Simple-All-Valid`. The nonzero full-sample difference between the two Qwen interfaces is therefore concentrated in the `Complex` subset.

The complete pairwise-AUC and Functional Coverage orderings nevertheless remain different within `Simple`, even though the two Qwen interfaces no longer exhibit a nonzero coverage difference there. This distinction shows that removing multi-unit, partial-overlap, and unresolved-validity cases does not by itself make pairwise ranking and set-level coverage orderings coincide.

Only 6 of the 24 `Simple` queries and 3 of the 17 `Simple-All-Valid` queries have selection space under budget three. The subset comparisons should therefore be interpreted in light of the limited number of queries on which the choice of three retained responses can affect Functional Coverage.

The subset definitions, sample counts, and complete numerical summaries used for this post-hoc comparison are reported in this appendix. The accompanying `natural50-posthoc-subsets` workflow reproduces the paired comparison and subset results from the frozen inputs and plan. It is separate from `natural50-final`, which reproduces the frozen primary analysis at budget three (Appendix G).