How Order-Sensitive Are LLMs?
OrderProbe for Deterministic Structural Reconstruction

Anonymous ACL submission

Abstract

Large language models (LLMs) excel at semantic understanding, yet their ability to reconstruct internal structure from scrambled inputs remains underexplored. Sentence-level restoration is difficult to evaluate automatically because scrambled sentences often admit multiple valid reorderings. We introduce **OrderProbe**, a deterministic benchmark for structural reconstruction using fixed four-character expressions in Chinese, Japanese, and Korean, which have a unique canonical order and thus support exact-match scoring. We further propose a diagnostic framework that evaluates models beyond recovery accuracy, including semantic fidelity, logical validity, consistency, robustness sensitivity, and information density. Experiments on twelve widely used LLMs show that structural reconstruction remains difficult even for frontier systems: zero-shot recovery frequently falls below 35%. We also observe a consistent gap between meaning-oriented generation and exact structural reconstruction, suggesting that structural robustness is not an automatic byproduct of semantic competence.

1 Introduction

Large language models (LLMs) have demonstrated strong performance in semantic understanding and text generation, yet their robustness to disruptions in internal structure remains unclear (OpenAI et al., 2024; DeepSeek-AI et al., 2025). Humans can often recover meaning from structurally disordered inputs with little effort, but evaluating this ability in machine models is difficult. At the sentence level, restoration is inherently ambiguous because a scrambled sentence may admit multiple valid reorderings. This makes deterministic evaluation difficult and blurs the distinction between structural competence and general fluency.

Existing robustness benchmarks mainly focus on surface noise or external perturbations such as instruction manipulation (Wang et al., 2021; Ribeiro et al., 2020; Li et al., 2024). Although these benchmarks have advanced robustness evaluation, they do not directly test whether a model can reconstruct a uniquely constrained sequence from scrambled components. This limitation is particularly relevant given evidence that Transformer models can exhibit limited sensitivity to word order (Sinha et al., 2021) and may fail to generalize structural relations under reversed or reordered contexts (Berglund et al., 2024).

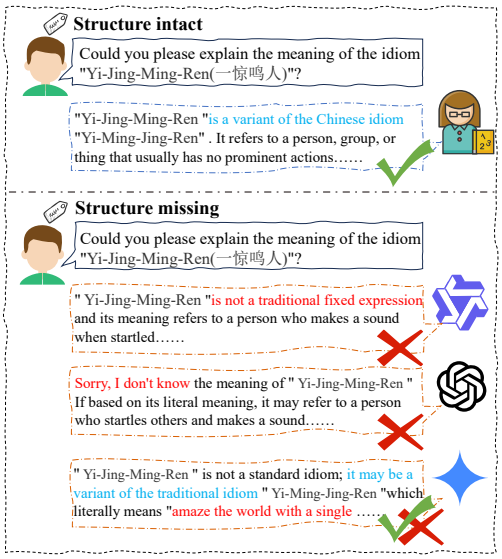

Figure 1: **OrderProbe example.** Humans recover the canonical order using semantic anchors, while LLMs often generate correct meanings but fail exact reconstruction.

To enable deterministic evaluation, we adopt fixed four-character expressions in Chinese, Japanese, and Korean as the target unit. Unlike scrambled sentences, such expressions typically have a single conventional order, making exact-match evaluation possible. We therefore study *structural reconstruction*: recovering the canonical internal order of lexicalized expressions under controlled permutations, rather than interpreting arbitrary character combinations. This setting differs from existing idiom and figurative language

Benchmark	Year	Language	Target Unit	Perturbation Level	Core Task
General Robustness Benchmarks					
RoTBench (Ye et al., 2024)	2024	En	Tool-use I/O	Real-world Noise	Tool-use Robustness (Tool Learning)
Instruct-Robust (Li et al., 2024)	2024	En	Instructions	External Injection	Instruction-Following Robustness
ReDial (Lin et al., 2025)	2025	En (Dialect)	Logic Problems	Dialectal Variation	Reasoning (Math/Logic)
Idiom & Figurative Language Benchmarks					
MMFLD (Lai et al., 2023)	2023	Multi	Fig. Language	*None* (Static)	Detection / Class.
WenMind (Cao et al., 2024)	2024	Zh	Classical Text	*None* (Static)	QA & Generation
Chengyu-Bench (Fu et al., 2025)	2025	Zh	Idioms	*None* (Static)	Understanding & Use
MChIRC (Wang et al., 2025b)	2025	Zh	Idiom-in-Context	*None* (Static)	Multimodal RC / Cloze
Idiom-Trans (Donthi et al., 2025)	2025	Multi	Idioms	*None* (Static)	Translation
OrderProbe (Ours)	**2025**	**CJK**	**Idioms**	**Internal Structure**	**Reconstruction**

Table 1: **Positioning of OrderProbe.** Existing robustness benchmarks mainly test surface-level noise or instruction-level perturbations, while idiom benchmarks treat expressions as static retrieval or transformation targets. OrderProbe uniquely evaluates deterministic internal-order reconstruction with exact-match scoring.

benchmarks, which mainly evaluate retrieval, interpretation, translation, or usage under canonical inputs (Zheng et al., 2019; Lai et al., 2023; Fu et al., 2025; Donthi et al., 2025; Cao et al., 2024).

Figure 1 illustrates the core challenge. Humans can often recover the canonical order from scrambled characters, whereas LLMs frequently produce meaning-aligned explanations while failing exact reconstruction. This reveals a gap between semantic access and exact structural recovery.

To study this phenomenon systematically, we introduce **OrderProbe**, a benchmark for evaluating structural reconstruction in Chinese, Japanese, and Korean. The dataset is built through multi-source collection, expert filtering, and hybrid semantic annotation to ensure linguistic reliability. It contains 3,543 curated expressions spanning six syntactic categories and multiple script types. We further propose a diagnostic evaluation framework that goes beyond recovery accuracy by measuring semantic fidelity, logical validity, structural consistency, robustness sensitivity, and information density. This enables fine-grained analysis of failure patterns rather than simple leaderboard comparison.

Our contributions are summarized as follows.

- We identify the ambiguity of sentence-level restoration and propose fixed four-character expressions as a deterministic proxy for evaluating structural reconstruction.
- We introduce OrderProbe, a multilingual benchmark containing 3,543 curated expressions across six syntactic categories.
- We propose a diagnostic framework that analyzes robustness, semantic fidelity, and generation errors beyond simple accuracy metrics.
- We evaluate twelve widely used LLMs on OrderProbe and show that exact reconstruction remains difficult, with zero-shot recovery often below 35% even for strong systems.

2 OrderProbe

OrderProbe is constructed through a controlled pipeline that transforms raw lexical resources into a deterministic benchmark for structural reconstruction. The pipeline consists of four stages: multi-source collection, expert filtering, semantic reference construction, and permutation-based perturbation generation. This section describes each stage and summarizes the resulting dataset.

2.1 Stage I: Multi-source Collection

We first assemble a candidate pool $\mathcal{C}_{raw}$ by aggregating four-character expressions from multiple publicly available resources, including lexical dictionaries and digital repositories. Each candidate item is associated with a script typology $T \in \{\text{ZH-CN}, \text{ZH-TW}, \text{JA}, \text{KO}\}$. The goal of this stage is to maximize coverage and diversity before quality control. This process yields approximately 4,000 raw candidates.

2.2 Stage II: Expert Filtering

To obtain a linguistically reliable core set, we apply an expert filtering function $\mathcal{F}$ that maps $\mathcal{C}_{raw}$ to a refined set $\mathcal{X}$:

$$\mathcal{X} = \mathcal{F}(\mathcal{C}_{raw}). \tag{1}$$

Filtering is performed by senior linguists and trained annotators with proficiency in the target languages. Annotations are conducted independently by five native-speaker annotators for each language, and final decisions are made by majority vote, followed by a brief consensus review for disputed cases. Items are removed if they are non-standard,

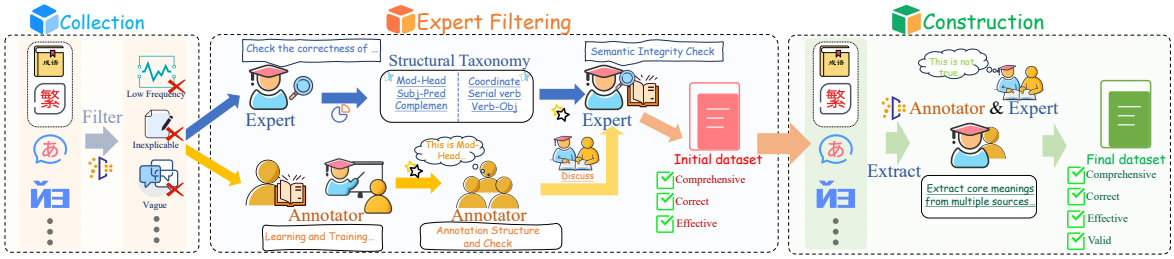

Figure 2: **OrderProbe construction pipeline.** We collect raw four-character expressions from multiple sources, apply expert filtering to obtain canonical roots, build semantic references via dictionary grounding and verified augmentation, and generate the full non-identity permutation space as structurally perturbed evaluation inputs.

ambiguous, modern internet slang, or exhibit multiple competing canonical forms. Expressions whose permutations could yield another conventional lexicalized entry are also removed to ensure a unique canonical target under the reconstruction objective. Inter-annotator agreement reaches substantial consistency (Fleiss' $\kappa = 0.835$). This process yields exactly 3,543 canonical expressions in $\mathcal{X}$, which form the benchmark roots.

2.3 Stage III: Semantic Reference Construction

For each canonical expression $x \in \mathcal{X}$, we construct a semantic reference set $\mathcal{S}_x$ to support evaluation beyond exact recovery. We retrieve dictionary definitions d_{dict} as the primary semantic anchor, following common practice in idiom evaluation (Zheng et al., 2019; Fu et al., 2025). To reduce brittleness to surface phrasing, we augment each definition with paraphrastic variants generated by LLMs and verified by annotators, similar to recent benchmark construction protocols (Li et al., 2024). The final reference set is defined as:

$$\mathcal{S}_x = \{d_{dict}\} \cup \mathcal{D}_{aug}. \tag{2}$$

This hybrid construction enables robust semantic and logical evaluation without translating across languages.

2.4 Stage IV: Permutation-based Perturbation Generation

The benchmark evaluates whether models can reconstruct the canonical order of an expression from scrambled constituents. Each canonical expression x is represented as a token sequence (t_1, t_2, t_3, t_4) with a fixed order. We apply a perturbation operator $\mathcal{P}$ that generates all non-identity permutations:

$$\mathcal{P}(x) = \{\pi(x) \mid \pi \in S_4, \pi(x) \neq x\}, \tag{3}$$

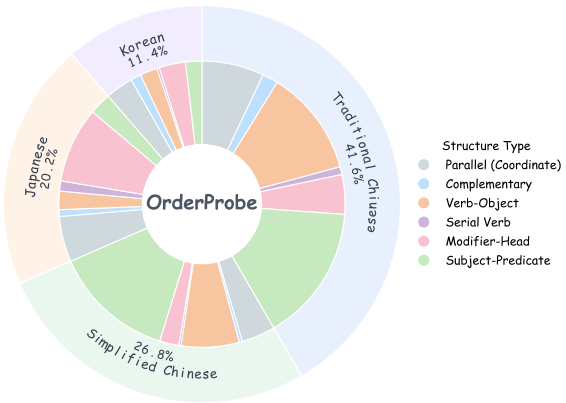

Figure 3: **Overview of OrderProbe.** The chart displays the distribution of script typologies and syntactic structures across the 3543 samples.

where S_4 denotes the permutation group over four elements. This yields 23 perturbed inputs per canonical expression, forming a complete combinatorial space for internal reordering. Unlike surface perturbations such as typos or synonym substitutions (Wang et al., 2021; Ribeiro et al., 2020), this operator preserves all lexical content while disrupting only the internal structure.

2.5 Dataset Statistics

The final dataset $\mathcal{D}$ consists of all perturbed variants generated from the canonical set $\mathcal{X}$. It covers four script settings (Simplified Chinese, Traditional Chinese, Japanese, and Korean) and six syntactic categories. The distribution of script typologies and structural patterns is shown in Figure 3. This design supports controlled analyses of how reconstruction varies across writing systems and syntactic templates. We treat Korean Hangul as a phonogrammatic control: syllable-block scrambling preserves syllable composition but provides weaker standalone semantic anchors than logographic scripts.

Category	Metric	Linguistic Definition & Rationale
Primary	**Recovery Rate**	Exact match of canonical order x. Measures global structural integrity.
Diagnostics	S_{Rob}	Robustness score based on degradation under S_n permutations (MDR/MDA).
	S_{Acc}^{mean}	Semantic fidelity via metaphorical alignment using Cross-Encoders and BERTScore.
	S_{Logic}	NLI-based entailment check between model output e and reference meaning.
	S_{Cons}	Stability across permutation variants, capturing invariance to internal shuffles.
	S_{Info}	Information density penalizing verbosity while rewarding high information-to-token ratio.

Table 2: **Evaluation Metrics on OrderProbe.** We prioritize Recovery Rate as the global structural indicator. Other diagnostic metrics characterize robustness, semantic fidelity, logical validity, invariance, and information density.

3 Evaluation Framework

Beyond deterministic exact-match reconstruction, OrderProbe introduces a multi-dimensional diagnostic framework that separates structural reconstruction from meaning-oriented generation. A single aggregate score is often insufficient: models may fail recovery due to hallucinated retrieval, positional sensitivity, format mimicry, or verbose "knowledge dumping," all of which can mask the underlying failure mechanism. We therefore design six complementary metrics(Table 2) computed over the full permutation space, enabling fine-grained failure-mode analysis.

Evaluation Scope. For each canonical expression x, we evaluate model behavior under all 23 non-identity permutations $\mathcal{P}(x)$, yielding $3543 \times 23 = 81{,}489$ perturbed inputs. Each output is scored by exact reconstruction, semantic alignment, and logical entailment, and then aggregated across permutations to quantify sensitivity and invariance. This produces a six-dimensional diagnostic signature per model, rather than only a leaderboard rank.

3.1 Primary Metric: Recovery Rate

The Recovery Rate is the global indicator of structural integrity:

$$\text{Recovery} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{x}_i = x_i). \tag{4}$$

Unlike sentence restoration (ill-posed due to multiple valid reorderings), four-character expressions have a unique canonical form, enabling deterministic evaluation.

3.2 Diagnostic Metrics (Beyond Recovery)

We introduce five diagnostic dimensions to explain *why* recovery fails.

Semantic Fidelity (S_{Acc}^{mean}). We evaluate explanation quality using a tiered hybrid semantic metric integrating cross-encoder relevance, multilingual embedding similarity, and lexical safeguards against entity-level hallucination (Appendix A.2).

Logical Validity (S_{Logic}). To detect fluent but contradictory definitions, we compute entailment probability between generated and reference meanings using a multilingual NLI model:

$$S_{\text{Logic}} = P_{\text{NLI}}(e \Rightarrow r). \tag{5}$$

Structural Consistency (S_{Cons}). Structural consistency measures whether a model preserves similar semantic behavior across different permutations of the same canonical expression. For each expression x, we compute the semantic score of the model output under every perturbed input in $\mathcal{P}(x)$ and summarize the dispersion of these scores across permutations. We define:

$$S_{\text{Cons}} = (1 - E_{\text{perf}}) \cdot (1 - R_{\text{sens}}), \tag{6}$$

where E_{perf} captures the average deviation from the best-performing permutation and R_{sens} captures the worst-case deviation. This metric penalizes both global instability and localized brittleness under internal reordering (Appendix A.3).

Robustness (S_{Rob}). We quantify robustness as degradation relative to the canonical baseline and decompose it into sequential robustness (permutation sensitivity) and structural robustness (variance across syntactic categories), combined via harmonic mean (Appendix A.5).

Information Density (S_{Info}). To penalize verbosity-based metric inflation, we introduce an information density score using a brevity penalty that rewards informative explanations per token (Appendix A.7).

Full formulations and implementation details are provided in Appendix A. We additionally test the robustness of the diagnostic pipeline under controlled substitutions of major components, including the cross-encoder, embedding similarity, NLI model, aggregation strategy, and weight scheme. Model rankings remain stable across variants, with Spearman correlations ranging from 0.842 to 0.931

Model Family	Setting	Diagnostic Component Metrics					Main Metric
		S_{Acc}^{mean}	S_{Cons}	S_{Logic}	S_{Rob}	S_{Info}	Recovery. (%)
Tier 1: High-Performance Reasoning Models							
Qwen-3-14B	Zero-shot	0.493	0.591	0.672	0.621	**0.579**	31.062
	+CoT	0.429	0.741	0.811	0.603	0.536	31.527 (↑0.465)
Qwen-3-VLThink	Zero-shot	**0.520**	0.539	0.712	0.645	0.520	30.928
	+CoT	0.471	0.716	0.888	0.632	0.533	48.376 (↑17.448)
Gemini-2.5-Flash	Zero-shot	0.493	0.621	0.646	0.631	0.540	30.487
	+CoT	0.455	0.738	0.738	**0.647**	0.246	41.559 (↑11.072)
Tier 2: Mid-Range General Models							
GPT-4o	Zero-shot	0.463	0.663	0.591	0.616	0.489	23.254
	+CoT	0.431	0.769	0.774	0.622	0.400	37.687 (↑14.433)
GLM-4VThink	Zero-shot	0.446	0.565	0.629	0.591	0.446	21.636
	+CoT	0.413	0.785	0.756	0.594	0.361	30.079 (↑8.443)
DeepSeek-V3.2	Zero-shot	0.472	0.593	0.575	0.609	0.444	20.104
	+CoT	0.482	0.727	**0.948**	0.643	**0.689**	**62.238** (↑42.134)
Tier 3: Efficient & Baseline Models							
DeepSeek-R1	Zero-shot	0.446	0.682	0.465	0.593	0.469	18.596
	+CoT	0.420	0.790	0.739	0.612	0.470	27.729 (↑9.133)
Claude-HaiKu-4.5	Zero-shot	0.448	0.656	0.494	0.612	0.466	18.017
	+CoT	0.398	0.714	0.652	0.551	0.293	21.646 (↑3.629)
Gemma-3-4B	Zero-shot	0.439	0.654	0.578	0.610	0.291	17.094
	+CoT	0.402	0.783	0.655	0.600	0.104	18.027 (↑0.933)
Llama-3-8B	Zero-shot	0.396	0.706	0.399	0.590	0.304	12.167
	+CoT	0.385	**0.815**	0.641	0.586	0.226	24.428 (↑12.261)
Llama-2-7B	Zero-shot	0.411	0.731	0.502	0.588	0.182	12.576
	+CoT	0.412	0.744	0.469	0.602	0.258	18.747 (↑6.171)
Qwen-3-8B	Zero-shot	0.446	0.617	0.487	0.604	0.403	15.786
	+CoT	0.392	0.715	0.492	0.580	0.269	12.594 (↓3.192)

Table 3: Comparative Assessment of Zero Shot and Chain of Thought Prompting across 12 LLMs on OrderProbe. This table reports performance metrics encompassing semantic accuracy, consistency and structural recovery.

relative to the original configuration, suggesting that the main conclusions are not driven by a particular metric choice (Appendix A.8).

4 Experiments

4.1 Experimental Setup

We evaluate all models on the full OrderProbe benchmark containing 3,543 canonical four-character expressions across four script settings (ZH-CN, ZH-TW, JA, KO). Given each scrambled input, models are required to output *(i)* the recovered canonical expression (a single four-character sequence) and *(ii)* a one-line semantic explanation in the corresponding language. All outputs are evaluated by exact-match reconstruction and our diagnostic metrics described in Section 3.

Models. We evaluate twelve widely used LLMs spanning open- and closed-source families. Open-source models include instruction-tuned checkpoints from Qwen (Qwen et al., 2025), DeepSeek (DeepSeek-AI et al., 2025), Llama (Grattafiori et al., 2024), and Gemma, run locally using official weights. Closed-source models include GPT (OpenAI et al., 2024), Gemini, Claude (Anthropic, 2024), and GLM (Team et al., 2025), accessed via official APIs with provider-default settings during Oct 2025–Jan 2026. All checkpoint and deployment details are provided in Appendix A.12.

Prompting Strategy and Output Constraints. We use a unified two-message prompting format (system + user) with strict schema enforcement to minimize format drift. We adopt language-matched prompting: each script setting is evaluated using prompts written in the same language as the input, and models are instructed to produce both the recovered expression and explanation in that language. We compare three settings: **(i)** Zero-shot (one-line output), **(ii)** Chain-of-Thought (CoT; exactly two lines: reasoning + final answer), and **(iii)** three-shot in-context learning (few-shot; two-line output following exemplars). All prompt templates are provided in Appendix D.

4.2 Overall Results and Analysis

Table 3 summarizes the benchmark results across twelve models under both Zero-shot and CoT settings. Across model families, we observe a consistent gap between meaning-oriented generation and exact structural recovery: models can often generate fluent, semantically plausible explanations via lexical retrieval from partially preserved cues, while still failing to reconstruct the canonical order

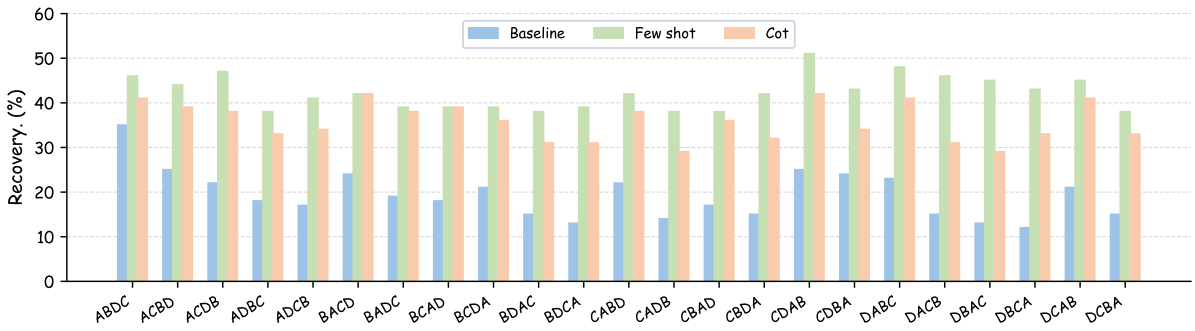

Figure 4: **Performance Comparison across 23 Permutation Patterns.** The chart illustrates recovery rates under baseline, few shot, and chain of thought settings, revealing the impact of anchor displacement on model stability.

Language	Component Metrics					Main Metric
	S_{Acc}^{mean}	S_{Cons}	S_{Logic}	S_{Rob}	S_{Info}	Recov. (%)
Simplified Chinese	**0.4817**	0.6445	**0.6854**	**0.6193**	0.4631	25.20
Traditional Chinese	0.4715	0.6184	0.6663	0.5998	0.4040	20.32
Japanese	0.4602	0.5897	0.5278	0.6048	**0.5388**	19.66
Korean	0.3990	**0.7625**	0.3014	0.5857	0.3906	5.72

Table 4: **Comparison across script settings.** Logographic scripts (Chinese/Japanese) yield substantially higher reconstruction accuracy than Korean Hangul, which serves as a phonogrammatic control condition. Component metrics are reported for diagnostic analysis alongside the primary Recovery score.

under permutation noise.

Reconstruction remains challenging. Exact reconstruction remains difficult even for strong systems, with zero-shot recovery frequently below 35%. This suggests that structural recovery is not an automatic byproduct of semantic competence.

CoT helps, but gains vary widely. CoT prompting often improves recovery, suggesting that explicit intermediate reasoning can encourage constraint checking before generation. However, the gains are strongly model-dependent and can reverse under unstable decoding.

Consistency is not correctness. High structural consistency can coincide with low recovery, as models may produce stable but generic explanations across permutations. This reinforces the need to interpret diagnostic metrics jointly.

Summary. Overall, OrderProbe shows that models can retrieve meanings or provide plausible definitions while still failing exact canonical reconstruction.

4.3 Permutation Sensitivity and Prompting Effects

Figure 4 compares recovery trends across all 23 permutation configurations under baseline, CoT, and few-shot settings. We observe strong dependence on anchor displacement: permutations that preserve semantically informative characters near their canonical neighborhoods yield higher recovery, while dispersed anchors significantly degrade performance.

Few-shot prompting yields the most uniform improvements across permutations, suggesting that explicit exemplars constrain decoding more effectively than generic reasoning traces. In contrast, CoT gains are less stable and may amplify format drift in weaker models.

4.4 Impact of Script Typology

Table 4 compares performance across the four script settings. We observe that reconstruction difficulty is strongly shaped by script typology.

Logographic scripts provide stronger local anchors. Chinese and Japanese exhibit substantially higher recovery than Korean. Logographic characters preserve standalone semantic cues under scrambling, providing local anchors that facilitate global reassembly.

Hangul is a phonogrammatic control. Korean Hangul behaves differently: syllable-block scrambling preserves syllable composition but weakens standalone semantic anchors. As a result, recovery collapses sharply, suggesting that reconstruction depends not only on positional decoding but also on interpretable local anchors.

Consistency inflation under extreme ambiguity. Notably, Korean yields high consistency despite low recovery. Under severe uncertainty, models often default to generic, stable explanations across permutations. This yields artificially high invariance but does not reflect true reconstruction ability, reinforcing the need to interpret diagnostic metrics jointly.

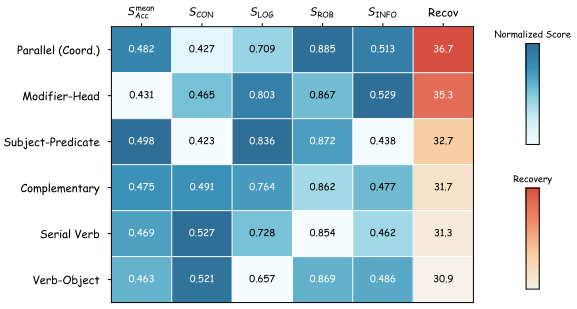

Figure 5: **Performance by syntactic category.** Parallel structures are easiest due to redundancy and symmetry cues, while asymmetric dependency patterns remain harder for reconstruction. Metrics are reported to diagnose whether failures arise from semantic gaps, instability, or structural sensitivity.

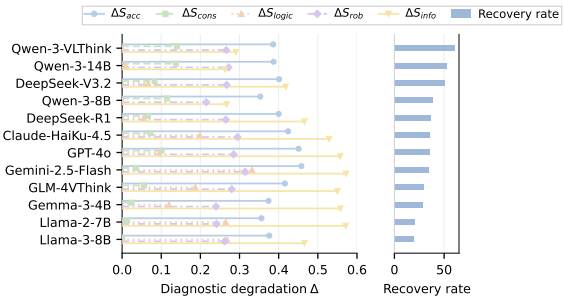

Figure 6: **Degradation under perturbed 3-shot inputs.** Left: per-metric drops ΔS_{acc}, ΔS_{cons}, ΔS_{logic}, ΔS_{rob}, ΔS_{info}, computed as *unperturbed* minus *perturbed 3-shot* under matched inference settings. Right: recovery rate for perturbed 3-shot inputs. Models are ordered by recovery rate (higher is more robust).

4.5 Analysis of Syntactic Patterns

Figure 5 reports results across six syntactic categories, revealing systematic structural variation beyond surface shuffling.

Parallel structures are easiest. Parallel and coordinate patterns are generally easier due to redundancy and symmetric templates. These cues effectively reduce the search space and provide ordering heuristics.

Meaning does not imply correct ordering. For subject–predicate patterns, models often recover the underlying proposition but fail to linearize characters into the precise canonical order, confirming that semantic access does not guarantee successful constraint-based sequence reconstruction.

Asymmetric dependencies remain difficult. Verb–object patterns are harder because they rely on directional dependencies that are disrupted by permutation noise, leading to semantically plausible but structurally incorrect guesses.

Input (Scrambled): 城 (City), 倾 (Topple), 国 (State), 倾 (Topple)
Target: 倾国倾城(Qing Guo Qing Cheng). Meaning: Extreme female beauty.

Qwen-3-8B — **Literal Interpretation**

Generated explanation:
指国家和城池被攻破或毁灭，常用于形容重大失败或灾难性后果。*(Refers to a country or city being breached.)*

Error note: The model produces a literal compositional interpretation based on individual characters rather than retrieving the conventional idiomatic meaning. Although the explanation is locally plausible, it does not correspond to the canonical lexicalized expression.

DeepSeek-V3 — **Precise Retrieval**

Generated explanation:
原指君主过分迷恋女色而亡国。后形容女子容貌极美。*(Originally refers to a monarch losing his kingdom.)*

Error note: Correctly captures the idiomatic metaphorical meaning.

Figure 7: **Case Study on Literal Interpretation.** Qwen-3-8B produces a fluent but non-canonical literal interpretation, while DeepSeek-V3 correctly retrieves the idiomatic metaphor.

4.6 Analysis of In-Context Learning

Figure 6 reports results under three-shot prompting.

Perturbations reveal structured degradation. Across models, diagnostic drops concentrate on particular components rather than moving in lockstep, suggesting that perturbations disrupt specific capabilities rather than inducing uniform instability.

High recovery does not imply preserved diagnostics. Some models retain relatively strong recovery while showing clear drops on particular diagnostics, whereas low-recovery models typically decline across multiple dimensions.

4.7 Case Study: Literal Interpretation under Structural Perturbation

Figure 7 shows a representative case on *Qing Guo Qing Cheng*. Some models retrieve the idiomatic metaphor correctly, while others produce a plausible but non-canonical literal interpretation. This example illustrates why exact reconstruction and diagnostic metrics are both necessary.

4.8 Memorization and Frequency Analysis

A natural concern is whether reconstruction mainly reflects memorization of frequent expressions. We therefore conduct a pilot frequency-stratified analysis (Appendix C.4). Although recovery improves

with idiom frequency, failures remain common even in high-frequency bins, suggesting that lexical familiarity alone does not guarantee successful reconstruction.

We analyze the role of lexical familiarity using a frequency-stratified experiment under progressively disrupted semantic signals. The frequency advantage diminishes as more signals are removed (Table 13).

To further examine whether reconstruction performance mainly reflects memorized idiom retrieval, we construct a synthetic control set consisting of non-attested four-character expressions. For each synthetic item, a canonical order is defined and the same permutation protocol and evaluation procedure are applied.

Models continue to exhibit low reconstruction accuracy on synthetic expressions, although performance is slightly lower than on idiomatic items. This result suggests that lexicalized idiom knowledge can provide partial cues, but the difficulty observed in OrderProbe cannot be reduced to memorized idiom lookup alone. Detailed construction procedures and quantitative results are provided in Appendix C.6 and Table 16.

Prior work on typoglycemia shows that scrambled text can often preserve semantic accessibility even when exact internal order is disrupted (Yu et al., 2024). This mechanism is consistent with our observation that models frequently generate meaning-aligned explanations while still failing exact canonical recovery. At the same time, typoglycemia-style robustness is often driven by lexical activation from partially preserved surface form (Wang et al., 2025a), whereas our synthetic-control results show that reconstruction remains difficult even when lexicalized idiom knowledge is reduced. This suggests that OrderProbe probes a stricter setting in which semantic access may survive scrambling, but exact canonical recovery still requires constraint-based reconstruction beyond lexical activation alone.

5 Related Work

Prior robustness benchmarks have examined lexical noise, adversarial perturbations, instruction manipulation, and reasoning under altered inputs. Early resources such as TextFlint (Wang et al., 2021) and CheckList (Ribeiro et al., 2020) mainly focus on surface-level variation, while later work extends robustness evaluation to instruction following (Li et al., 2024) and logical reasoning (Lin et al., 2025). Related studies on typoglycemia and scrambled text ask whether models can still recognize or understand perturbed inputs. In contrast, OrderProbe does not test robustness to generic noise, but exact reconstruction of a unique canonical form from internally permuted components.

Research on idiom and figurative language processing has largely treated idioms as static lexical units. Benchmarks such as ChID (Zheng et al., 2019), MMFLD (Lai et al., 2023), Chengyu-Bench (Fu et al., 2025), and Idiom-Trans (Donthi et al., 2025) mainly evaluate cloze prediction, interpretation, translation, or usage under canonical inputs. Classical-language benchmarks such as WenMind (Cao et al., 2024) similarly emphasize semantic understanding rather than recovery under internal perturbation. By contrast, OrderProbe focuses on whether models can recover the conventional internal order of lexicalized four-character expressions, enabling deterministic exact-match evaluation.

Our findings are also related to work on positional bias and structural limitations in Transformer-based models. Sinha et al. (2021) show that models can be unexpectedly insensitive to word order in some settings, and Berglund et al. (2024) document failures to generalize relational structure under reversal. OrderProbe complements these findings with a controlled benchmark in Chinese, Japanese, and Korean, showing that even when local semantic cues remain partially available, exact canonical reconstruction under internal permutation remains challenging for current LLMs.

6 Conclusion

We introduce **OrderProbe**, a deterministic benchmark for structural reconstruction in Chinese, Japanese, and Korean four-character expressions. Experiments on twelve LLMs show that exact recovery remains difficult, with zero-shot performance often below 35%, highlighting a gap between semantic access and exact structural reconstruction. We also find strong effects of script typology: logographic scripts offer stronger local anchors, whereas Hangul provides a phonogrammatic control with weaker local anchoring. OrderProbe is intended to complement broader evaluations of order sensitivity, and we release the benchmark and diagnostic framework to support future research on structural robustness in LLMs.

Limitations

OrderProbe evaluates structural reconstruction in a controlled lexical setting, but it does not cover the full range of linguistic structures. Our analysis is limited to six syntactic categories and to fixed four-character expressions in three languages. We also do not study alphabetic or morphologically rich systems in detail.

In addition, some expressions are likely to appear in pretraining corpora, so reconstruction may partly benefit from memorized lexical knowledge. Although our frequency analysis suggests that memorization alone does not explain the results, broader tests on novel, synthetic, or non-idiomatic controls remain for future work.

Ethics Statement

The OrderProbe dataset integrates linguistic resources from the public domain under fair use principles for academic research. We confirm that the collected expressions are free from copyright restrictions and strictly adhere to data usage policies. All participants provided informed consent and followed a rigorous protocol to exclude offensive or discriminatory content.

Our data construction pipeline utilizes LLMs for semantic augmentation in full compliance with provider terms of service. We declare that all generated outputs underwent human verification to ensure factual accuracy and safety. This work serves as a diagnostic benchmark for enhancing model robustness. We foresee no direct negative societal impacts or misuse risks associated with the release of this dataset.

References

Anthropic. 2024. Claude model report. https://www.anthropic.com/transparency/model-report. Accessed: 2025-01.

Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Stickland, Tomasz Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on "a is b" fail to learn "b is a". In *I Can't Believe It's Not Better Workshop: Failure Modes in the Age of Foundation Models*.

Jiahuan Cao, Yang Liu, Yongxin Shi, Kai Ding, and Lianwen Jin. 2024. Wenmind: A comprehensive benchmark for evaluating large language models in chinese classical literature and language arts. In *Advances in Neural Information Processing Systems*, volume 37, pages 51358–51410. Curran Associates, Inc.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2025. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Sundesh Donthi, Maximilian Spencer, Om B. Patel, Joon Young Doh, Eid Rodan, Kevin Zhu, and Sean O'Brien. 2025. Improving LLM abilities in idiomatic translation. In *Proceedings of the First Workshop on Language Models for Low-Resource Languages*, pages 175–181, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yicheng Fu, Zhemin Huang, Liuxin Yang, Yumeng Lu, and Zhongdongming Dai. 2025. Chengyu-bench: Benchmarking large language models for chinese idiom understanding and use. *Preprint*, arXiv:2506.18105.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Huiyuan Lai, Antonio Toral, and Malvina Nissim. 2023. Multilingual multi-figurative language detection. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9254–9267, Toronto, Canada. Association for Computational Linguistics.

Zekun Li, Baolin Peng, Pengcheng He, and Xifeng Yan. 2024. Evaluating the instruction-following robustness of large language models to prompt injection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 557–568, Miami, Florida, USA. Association for Computational Linguistics.

Fangru Lin, Shaoguang Mao, Emanuele La Malfa, Valentin Hofmann, Adrian de Wynter, Xun Wang, Si-Qing Chen, Michael J. Wooldridge, Janet B. Pierrehumbert, and Furu Wei. 2025. Assessing dialect fairness and robustness of large language models in reasoning tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6317–6342, Vienna, Austria. Association for Computational Linguistics.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Marco Tulio Ribeiro, Tongshuang Wu, Carlos Guestrin, and Sameer Singh. 2020. Beyond accuracy: Behavioral testing of NLP models with CheckList. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4902–4912, Online. Association for Computational Linguistics.

Koustuv Sinha, Prasanna Parthasarathi, Joelle Pineau, and Adina Williams. 2021. UnNatural Language Inference. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7329–7346, Online. Association for Computational Linguistics.

5 Team, Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, Kedong Wang, Lucen Zhong, Mingdao Liu, Rui Lu, Shulin Cao, Xiaohan Zhang, Xuancheng Huang, Yao Wei, and 152 others. 2025. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *Preprint*, arXiv:2508.06471.

Chenxi Wang, Tianle Gu, Zhongyu Wei, Lang Gao, Zirui Song, and Xiuying Chen. 2025a. Word form matters: Llms' semantic reconstruction under typoglycemia. *Preprint*, arXiv:2503.01714.

Tongguan Wang, Mingmin Wu, Guixin Su, Dongyu Su, Yuxue Hu, Zhongqiang Huang, and Ying Sha. 2025b. Mchirc: A multimodal benchmark for chinese idiom reading comprehension. In *AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA*, pages 25398–25406. AAAI Press.

Xiao Wang, Qin Liu, Tao Gui, Qi Zhang, Yicheng Zou, Xin Zhou, Jiacheng Ye, Yongxin Zhang, Rui Zheng, Zexiong Pang, Qinzhuo Wu, Zhengyan Li, Chong Zhang, Ruotian Ma, Zichu Fei, Ruijian Cai, Jun Zhao, Xingwu Hu, Zhiheng Yan, and 15 others. 2021. TextFlint: Unified multilingual robustness evaluation toolkit for natural language processing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 347–355, Online. Association for Computational Linguistics.

Junjie Ye, Yilong Wu, Songyang Gao, Caishuang Huang, Sixian Li, Guanyu Li, Xiaoran Fan, Qi Zhang, Tao Gui, and Xuanjing Huang. 2024. Rotbench: A multi-level benchmark for evaluating the robustness of large language models in tool learning. *Preprint*, arXiv:2401.08326.

Miao Yu, Junyuan Mao, Guibin Zhang, Jingheng Ye, Junfeng Fang, Aoxiao Zhong, Yang Liu, Yuxuan Liang, Kun Wang, and Qingsong Wen. 2024. Mind scramble: Unveiling large language model psychology via typoglycemia. *Preprint*, arXiv:2410.01677.

Chujie Zheng, Minlie Huang, and Aixin Sun. 2019. ChID: A large-scale Chinese IDiom dataset for cloze test. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 778–787, Florence, Italy. Association for Computational Linguistics.

A Formal Definitions and Rationale of Evaluation Metrics

This appendix details the mathematical formulation and design rationale for the evaluation metrics employed in OrderProbe. To avoid the ambiguity often introduced by aggregated leaderboards, we adopt a **multi-dimensional diagnostic framework**. Each metric is designed to capture a specific failure pattern—ranging from structural rigidity to semantic hallucination—thereby providing a granular dissection of model capabilities. These metrics are reported independently in the main paper and are not aggregated into a single score.

A.1 Primary Metric: Recovery Rate

Rationale The ability to recover the canonical form serves as a direct indicator of structural reconstruction ability. Unlike semantic comprehension, which can be inferred probabilistically from keywords, structural reconstruction requires resolving specific positional dependencies. Strict matching is necessary to distinguish exact canonical reconstruction from partial meaning recovery.

Formulation Let x denote a canonical four-character idiom and $x' = \mathcal{T}(x, \pi)$ its internally reordered variant. Given a model output o, reconstruction is marked as correct only if the recovered expression exactly matches the canonical idiom x. Partial matches, paraphrases, or explanations without exact reconstruction are not counted as correct.

The Recovery Rate is defined as:

$$\text{Recovery} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(\hat{x}_i = x_i), \tag{7}$$

In this context, $\mathbb{I}(\cdot)$ represents the indicator function.

A.2 Semantic Accuracy

Rationale and Component Selection Assessing idiom interpretation requires capturing both deep

metaphorical meaning and precise entity details. Single-dimensional metrics cannot capture this duality. Vector-based similarity measures often miss specific entity errors, while n-gram overlap fails to recognize valid metaphorical paraphrases. To address this, we construct a tiered hybrid metric, S_{Acc}, that integrates signals from both interaction-based and lexical levels.

Weighting Justification We assign weights to reflect the functional hierarchy of each verification method. We treat the cross-encoder as the primary evaluator and assign it a weight of 0.5. Unlike bi-encoder architectures, which compress sentences into independent vectors, cross-encoders perform full self-attention over the concatenated input. This enables them to detect subtle semantic entailments that are essential for precise interpretation. We assign a moderate weight of 0.3 to an ensemble of embedding-based measures, including BERTScore, STS, and cosine similarity, which serves as a robust semantic baseline. Relying on a single embedding model may introduce architectural bias. Integrating distinct representation approaches reduces variance and provides a more general measure of semantic proximity. We allocate the remaining weight of 0.2 to the F_β score as a lexical safeguard. Models can exhibit high similarity scores while hallucinating or mismatching specific entities. This lexical component therefore penalizes factual discrepancies.

Formulation For an explanation e and a reference definition r, the composite Semantic Accuracy score is defined as:

$$S_{\text{Acc}} = w_1 \cdot S'_{\text{ce}} + w_2 \cdot \left(\frac{S'_{\text{bert}} + S'_{\text{sts}} + S'_{\text{cos}}}{3} \right) + w_3 \cdot S_{f\beta}. \tag{8}$$

S'_{ce} denotes the normalized score produced by the cross-encoder. The term in parentheses is the averaged embedding similarity score. $S_{f\beta}$ denotes the lexical F_β score. We set the weights to $w_1 = 0.5$, $w_2 = 0.3$, and $w_3 = 0.2$ based on the hierarchy described above.

A.3 Structural Consistency

Rationale Structural consistency evaluates whether a model preserves similar semantic behavior across different permutations of the same canonical expression. Since our evaluation uses deterministic decoding, each permuted input yields a single model output. We therefore measure consistency across the permutation set rather than across repeated generations of the same input.

Formulation Let $\mathcal{P}(x)$ denote the complete set of non-identity permutations for a canonical expression x. For each permutation $p \in \mathcal{P}(x)$, the model produces one output explanation, whose semantic accuracy score is denoted by $S_{\text{Acc}}(p)$.

We first define the best semantic performance observed across the permutation set:

$$S_{\max}(x) = \max_{p \in \mathcal{P}(x)} S_{\text{Acc}}(p). \tag{9}$$

We then define the average deviation from this best-performing permutation as:

$$E_{\text{perf}}(x) = \frac{1}{|\mathcal{P}(x)|} \sum_{p \in \mathcal{P}(x)} \left(S_{\max}(x) - S_{\text{Acc}}(p) \right). \tag{10}$$

We define the worst-case deviation as:

$$R_{\text{sens}}(x) = \max_{p \in \mathcal{P}(x)} \left(S_{\max}(x) - S_{\text{Acc}}(p) \right). \tag{11}$$

Finally, we compute structural consistency as:

$$S_{\text{Cons}}(x) = (1 - E_{\text{perf}}(x)) \times (1 - R_{\text{sens}}(x)). \tag{12}$$

This formulation assigns a high score to models whose semantic behavior remains stable across different internal reorderings of the same expression.

A.4 Degradation Under Internal Reordering

Rationale and Normalization Logic Isolating the impact of structural noise requires a performance baseline for each specific idiom. We use the score obtained from the canonical non-reordered input, S_{orig}, as the reference capability. A low score on a scrambled input indicates structural fragility only if the model demonstrates competence on the canonical form. Direct comparison without normalization fails to distinguish between structural sensitivity and general knowledge deficits. Consequently, we quantify degradation as a relative loss ratio. MDR captures the expected utility loss under random structural perturbations. MDA captures a lower bound on performance reliability, reflecting the worst-case structural collapse. The maximization term in MDA extracts the worst-case signal across the permutation group to maintain a consistent definition.

Formulation We formalize the degradation metrics through the following equations. $\mathcal{P}$ denotes the set of permutations applied to the idiom. S_{orig} represents the semantic accuracy score of the model on the canonical input. S^p_{pert} denotes the score under permutation p. We define the Mean Degradation Rate (MDR) as the average relative performance drop across all permutations.

$$\text{MDR} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \frac{S_{\text{orig}} - S^p_{\text{pert}}}{S_{\text{orig}}} \tag{13}$$

We define the Maximum Degradation Amount (MDA) as the maximum relative drop observed across the permutation group.

$$\text{MDA} = \max_{p \in \mathcal{P}} \left(\frac{S_{\text{orig}} - S^p_{\text{pert}}}{S_{\text{orig}}} \right) \tag{14}$$

These values quantify the average and worst-case loss of capability attributable to internal reordering.

A.5 Robustness Metrics

Rationale and Dual Dimensionality We assess robustness along two perturbation dimensions: Sequential Robustness and Structural Robustness. Sequential Robustness measures the resilience of the model against internal ordering permutations. Structural Robustness evaluates the stability of performance across distinct syntactic categories. We select the mean semantic accuracy $S^{\text{mean}}_{\text{Acc}}$ as the basis for these metrics. While maximum scores reflect peak recovery capability, average scores capture stability under stochastic degradation and thus provide a stricter and more sensitive indicator of robustness. The final metric integrates these dimensions using a harmonic mean, ensuring that high robustness requires resilience to both combinatorial scrambling and syntactic variation.

Sequential Robustness Formulation We quantify Sequential Robustness S_{seq} by combining the mean and maximum degradation rates defined in Appendix A.4. We define S_{seq} as the complement of a weighted sum of these degradations. We assign equal weights $\alpha = 0.5$ and $\beta = 0.5$ to balance average stability and limit-state resilience.

$$S_{\text{seq}} = 1 - (\alpha \cdot \text{MDR} + \beta \cdot \text{MDA}) \tag{15}$$

Structural Robustness Formulation We quantify Structural Robustness S_{struct} by measuring performance variance across syntactic categories. Let T denote the set of six syntactic structure types indexed by k. For the k-th structure, we compute the intra-group mean accuracy μ_k over the idiom set D_k.

$$\mu_k = \frac{1}{|D_k|} \sum_{x \in D_k} S^{\text{mean}}_{\text{Acc}}(x) \tag{16}$$

We calculate the standard deviation σ of these group means to capture the sensitivity of the model to syntactic form. We define S_{struct} as the complement of the normalized deviation, ensuring that consistent performance across diverse structures yields a high score.

$$S_{\text{struct}} = 1 - \text{Normalize}(\sigma(\mu_1, \mu_2, ..., \mu_6)) \tag{17}$$

Composite Robustness Score We compute the final Robustness Score S_{Rob} as the harmonic mean of the sequential and structural components. This aggregation penalizes imbalance, requiring the model to demonstrate robustness in both internal ordering and syntactic generalization.

$$S_{\text{Rob}} = \frac{2 \cdot S_{\text{seq}} \cdot S_{\text{struct}}}{S_{\text{seq}} + S_{\text{struct}}} \tag{18}$$

A.6 Logical Validity

Rationale and Entailment Check High lexical similarity does not guarantee factual correctness. Models may generate explanations that share keywords with the target while conveying contradictory meanings through negation or causal inversion. Lexical metrics fail to detect these logical errors. To address this, we employ a Natural Language Inference (NLI) model as a validity filter. This component evaluates whether the generated explanation logically entails the canonical definition, ensuring that high scores reflect true semantic understanding rather than superficial keyword matching.

Formulation We quantify logical validity S_{Logic} as the probability of entailment predicted by a pre-trained NLI model. Given a generated explanation e and a reference definition r, we compute the entailment probability:

$$S_{\text{Logic}} = P_{\text{NLI}}(e \Rightarrow r) \tag{19}$$

A.7 Information Density

Rationale and Verbosity Penalty Models frequently employ knowledge-dumping strategies, generating excessively verbose explanations to maximize keyword coverage. This behavior inflates recall-based metrics without reflecting precise understanding. To counteract this, we introduce an Information Density metric that rewards

high information content per token. We impose a brevity penalty (BP) on outputs that exceed the reference length without proportional semantic gain. This mechanism encourages the model to prioritize concise and accurate reasoning over redundant generation.

Formulation We compute Information Density S_{Info} by scaling the precision score with a length-based penalty:

$$S_{\text{Info}} = \text{BP} \times P_{\text{ROUGE}} \tag{20}$$

P_{ROUGE} denotes the ROUGE precision score. BP represents the brevity penalty factor, downweighting explanations that exhibit low information density relative to the canonical definition.

A.8 Robustness of Diagnostic Metrics

Across all ten variants, model rankings remain stable, with Spearman correlations ranging from 0.842 to 0.931 relative to the original pipeline. This stability suggests that the main empirical patterns reported in the paper are not an artifact of a particular metric configuration. In particular, substitutions of the cross-encoder backbone, embedding model, NLI model, aggregation strategy, and weight scheme do not alter the qualitative direction of the findings. These results support the robustness of the diagnostic framework under substantial changes to its scoring components.

A.9 Human Audit Protocol

We conduct a targeted human audit to validate the proposed multi-dimensional evaluation framework. The audit assesses whether automated evaluation dimensions align with expert judgments under explicit and controlled rating criteria.

For each language, we sample a fixed subset of instances from the evaluation set without conditioning on model performance or automated metric values. The audit includes 290 instances for ZH-CN, 430 instances for ZH-TW, 170 instances for JA, and 110 instances for KO. Each instance is independently annotated by five annotators who are proficient in the corresponding language. Annotators evaluate model-generated explanations without access to automated scores. Final human scores are obtained by averaging annotator ratings, and inter-annotator agreement is moderate to high across languages.

Human evaluation follows predefined rubrics that map qualitative judgments to normalized scores. All dimensions are rated independently.

Accuracy score s_{Acc}^{mean}. A score of at least 0.8 indicates a fully correct explanation. Scores between 0.6 and 0.8 indicate that the main meaning is correct with minor inaccuracies. Scores between 0.4 and 0.6 indicate noticeable errors despite a largely correct meaning. Scores below 0.4 indicate severe errors and an incorrect explanation.

Logical validity score s_{Logic}. A score of at least 0.8 indicates that the explanation is logically consistent with the reference meaning and does not introduce contradictions. Scores between 0.6 and 0.8 indicate that the main meaning is correct but includes minor logical omissions or slight misspecifications. Scores between 0.4 and 0.6 indicate mixed correctness, with partial contradictions or incorrect causal relations. Scores below 0.4 indicate a logically inconsistent or contradictory explanation.

Information score s_{Info}. A score of at least 0.8 indicates a concise and informative explanation with minimal redundancy. Scores between 0.6 and 0.8 indicate mostly informative explanations with some redundancy. Scores between 0.4 and 0.6 indicate limited information density or substantial redundancy. Scores below 0.4 indicate explanations that are either highly redundant or minimally informative.

To quantify alignment at the instance level, we compute Spearman's rank correlation between human audit ratings and automated metric scores on the same audited samples. Correlations are reported for all dimensions that produce per-instance scores under both human and automated evaluation. Table 6 summarizes the alignment results across languages.

A.10 Alignment Between Human Evaluation and Automated Metrics

This section examines the alignment between human expert judgments and the proposed automated multi-dimensional evaluation framework. Table 7 compares aggregated human audit scores with automated metric outputs across languages.

Across all evaluated languages, the automated metrics exhibit consistent directional alignment with human assessments. Dimensions related to logical coherence, robustness, and internal consistency show similar relative patterns under both evaluation methods, suggesting that the metrics capture structured properties of model behavior rather than superficial numerical differences.

Differences in absolute scores are observed be-

Variant	CE Model	CE Agg	Embedding	Sim Func	BERTScore	NLI Model	NLI Agg	Weight Scheme	ρ	Effect Dir.
Original	BGE-base	max	BGE-small / MiniLM	cosine	✓	mDeBERTa	bidir-max	fixed	—	+
V1	MiniLM CE	max	same	cosine	✓	mDeBERTa	bidir-max	fixed	0.917	+
V2	BGE-base	mean	same	cosine	✓	mDeBERTa	bidir-max	fixed	0.904	+
V3	BGE-base	max	distiluse	cosine	✓	mDeBERTa	bidir-max	fixed	0.873	+
V4	BGE-base	max	same	dot	✓	mDeBERTa	bidir-max	fixed	0.898	+
V5	BGE-base	max	same	cosine	✗	mDeBERTa	bidir-max	fixed	0.931	+
V6	BGE-base	max	same	cosine	✓	XLM-R NLI	bidir-max	fixed	0.897	+
V7	BGE-base	max	same	cosine	✓	mDeBERTa	unidir	fixed	0.872	+
V8	BGE-base	max	same	cosine	✓	mDeBERTa	bidir-mean	fixed	0.842	+
V9	BGE-base	max	same	cosine	✓	mDeBERTa	bidir-max	equal weights	0.893	+
V10	— (no CE)	—	same	cosine	✓	mDeBERTa	bidir-max	fixed	0.854	+

Table 5: **Metric stress tests under controlled substitutions.** We vary the cross-encoder backbone, aggregation strategy, embedding model, similarity function, BERTScore inclusion, NLI backbone, NLI aggregation, and weight scheme. ρ denotes Spearman correlation of model rankings with the original pipeline. The qualitative effect direction remains unchanged across all tested variants.

Language	N	ρ_{Acc}^{mean}	ρ_{Logic}	ρ_{Info}
ZH-CN	290	0.787	0.822	0.743
ZH-TW	430	0.754	0.801	0.717
JA	170	0.694	0.739	0.665
KO	110	0.723	0.762	0.685

Table 6: Instance level Spearman rank correlation between human audit ratings and automated metric scores on the audited samples.

tween human audits and automated evaluations. Such variation is expected given the interpretive nature of human judgment and the formal constraints of automated scoring. Importantly, the relative ordering of evaluation dimensions remains stable across methods, indicating that the proposed multi-dimensional framework preserves the structure of human evaluation across languages.

Language	Evaluation Method	s_{Acc}^{mean}	s_{Cons}	s_{Logic}	s_{Rob}	s_{Info}
ZH-CN	Human Audit	0.412	0.758	0.829	0.715	0.456
	Automated Metric	0.482	0.663	0.749	0.625	0.516
ZH-TW	Human Audit	0.405	0.731	0.779	0.701	0.336
	Automated Metric	0.480	0.641	0.699	0.613	0.406
JA	Human Audit	0.413	0.654	0.632	0.708	0.547
	Automated Metric	0.479	0.559	0.552	0.621	0.617
KO	Human Audit	0.353	0.875	0.443	0.698	0.329
	Automated Metric	0.413	0.788	0.363	0.604	0.419

Table 7: Comparison between human expert audits and automated multi dimensional evaluation scores across languages.

A.11 Cross-Lingual Metric Reliability

We further examine the reliability of automated diagnostic metrics across script settings. For each language, we compute Spearman correlation between human audit scores and automated scores, along with bootstrap confidence intervals. This analysis is conducted separately for semantic accuracy, logical validity, and information density, since these dimensions are directly rated by human annotators.

Metric reliability varies across languages. The Chinese settings show relatively stronger human–automatic alignment, while Japanese and Korean exhibit lower correlations. This may reflect weaker lexical overlap, different segmentation behavior, and reduced compatibility between multilingual semantic models and language-specific idiomatic meanings. Therefore, we interpret automated diagnostic scores as relative indicators rather than absolute semantic judgments, especially for script settings where human–automatic agreement is lower.

A.12 Generation Protocol and Nondeterminism

All models are evaluated using inference configurations consistent with those commonly adopted in large-scale benchmarking studies. For models that expose a temperature parameter, we set the temperature to 0 to enforce deterministic decoding. No task-specific tuning or customized decoding strategies are applied.

Open-source models are run locally using official checkpoints and standard inference pipelines. Closed-source models are accessed through their official APIs, following the default configurations recommended by their respective providers. All API-based evaluations are conducted between October 2025 and January 2026. When configurable parameters are available, we align them across models to the extent possible; otherwise, we retain provider defaults.

Across all experiments, we use identical prompt templates and input perturbations. No fine-tuning, reranking, or post-processing is performed. This deployment protocol ensures that performance dif-

ferences primarily reflect intrinsic model behaviors rather than artifacts of inference configuration, stochastic decoding, or serving infrastructure.

B Appendix: Details of Semantic and Logical Evaluation

This appendix specifies implementation details of the automated evaluation metrics for semantic explanation quality and logical consistency to ensure reproducibility and cross-lingual comparability.

B.1 Semantic Explanation Scoring

Semantic explanation quality is measured using a composite score that integrates interaction-based semantic matching, representation-based similarity, and content coverage.

For interaction-based matching, we use the cross-encoder reranker BAAI/bge-reranker-base. Given a model-generated explanation and a set of reference explanations, the cross-encoder produces a relevance score for each explanation–reference pair. These scores are normalized to a bounded scale and aggregated across multiple references using either max or mean aggregation. The same cross-encoder model is applied uniformly across all languages.

For representation-based similarity, we use sentence embedding models to compute cosine similarity between explanations and references. For Simplified Chinese and Traditional Chinese, we use BAAI/bge-small-zh-v1.5. For Japanese and Korean, we use sentence-transformers/paraphrase-multilingual-MiniLM-L12-v2. These models are selected for their multilingual coverage and efficiency. No translation is applied, and all texts are evaluated in their original language.

In addition, we optionally include BERTScore F1 as a token-level semantic similarity signal, using language-specific settings for Chinese, Japanese, and Korean. Lexical overlap features are treated as auxiliary signals and do not dominate the final score.

The final semantic accuracy score is computed as a weighted linear combination of cross-encoder similarity, representation-based similarity, and the content-word F_β score, with fixed weights shared across languages.

B.2 Logical Consistency via Natural Language Inference

Logical consistency between model explanations and reference explanations is measured using the multilingual natural language inference (NLI) classifier MoritzLaurer/mDeBERTa-v3-base-mnli-xnli. The same NLI model is applied to all languages without translation.

For each reference explanation, we compute NLI in two directions: from the reference to the model output and from the model output to the reference. This bidirectional evaluation reduces asymmetry caused by verbosity or paraphrasing. We use the entailment probability as the primary signal, while the contradiction probability serves as a secondary penalty when entailment evidence is present.

Specifically, for each reference, the score is defined as the maximum entailment probability across the two directions, with an additional contradiction-aware adjustment activated only when entailment exceeds a small threshold. Scores are aggregated across multiple references using max aggregation. This reflects the assumption that a model explanation is logically consistent if it entails at least one valid reference explanation.

The NLI scores are treated as continuous consistency signals and are not converted into binary decisions. No post-hoc calibration is applied, and scores are used only for relative comparison and averaging.

B.3 Reference Construction and Evaluation Scope

Reference explanations consist of canonical dictionary definitions augmented with paraphrased variants to cover legitimate surface variation. Dictionary definitions serve as the primary semantic anchors, while paraphrased references reduce sensitivity to wording differences. All evaluation metrics aggregate over multiple references to avoid dependence on any single phrasing.

We evaluate the main reconstruction accuracy metric alongside semantic and logical metrics. While some items may appear in pretraining corpora, reconstruction accuracy is not interpreted in isolation. Instead, differences across semantic, logical, and robustness metrics are used to assess structure-aware semantic grounding beyond surface recall.

C Additional Validation and Controls

This section provides additional validation analyses to calibrate the difficulty of OrderProbe and to examine the robustness of the reported findings. We focus on six complementary aspects: Korean

Model	Avg. Tokens	Frag. Rate (%)	Recovery (%)
Qwen-3-14B	4.31	22.4	11.8
Qwen-3-VLThink	4.28	21.7	12.6
DeepSeek-V3.2	4.47	25.1	14.3
GPT-4o	4.82	31.6	8.9
Gemini-2.5-Flash	4.76	29.8	9.4
GLM-4VThink	5.12	38.3	6.7
Llama-3-8B	6.84	64.5	3.8
Gemma-3-4B	6.21	57.2	4.6

Table 8: **Tokenizer fragmentation statistics for Korean Hangul syllable-block inputs.** Avg. Tokens denotes the average number of model-specific subword tokens used to encode a four-block Korean expression. Frag. Rate denotes the percentage of expressions encoded with more than four tokens.

processing and tokenization, human reconstruction baselines, uncertainty estimation, frequency-stratified analysis, synthetic-control evaluation, and cross-lingual metric reliability. These analyses are intended to support the main conclusion that exact structural reconstruction remains challenging even when semantic access is partially preserved.

C.1 Korean Processing and Perturbation Unit

For the Korean setting, we apply perturbations at the level of Hangul syllable blocks rather than decomposed jamo units. A Korean four-unit expression is represented as a sequence of written syllable blocks (h_1, h_2, h_3, h_4), and OrderProbe permutes these blocks in the same way as it permutes Chinese characters or Japanese written units.

This design has two implications. First, Korean perturbation preserves the internal composition of individual Hangul syllables and should not be interpreted as jamo-level scrambling. Second, compared with logographic Chinese characters and Japanese Kanji, Hangul syllable blocks generally provide weaker standalone semantic anchors under permutation. We therefore treat Korean as a phonogrammatic control condition in which local semantic cues are reduced, rather than as a condition where syllable composition itself is destroyed.

To further examine this issue, we compute tokenizer fragmentation statistics for Korean inputs. For each model, we record the number of model-specific subword tokens used to encode a four-block Korean expression and report the average token count and fragmentation rate. This analysis helps separate script-level effects from tokenizer-level artifacts.

Language	Items	Inputs	Human	Best LLM	Avg. LLM
ZH-CN	120	360	88.6	57.9	27.4
ZH-TW	120	360	84.1	50.8	23.5
JA	100	300	72.8	42.3	22.1
KO	100	300	46.5	16.9	7.2

Table 9: **Human reconstruction baseline.** Human participants recover the canonical expression from the same scrambled inputs used for model evaluation. Human, Best LLM, and Avg. LLM denote exact recovery rates (%).

Language	Recovery (%)	Time (s)	Confidence
ZH-CN	88.6	7.4	4.32
ZH-TW	84.1	8.1	4.18
JA	72.8	10.6	3.76
KO	46.5	14.9	3.08

Table 10: **Human reconstruction difficulty statistics.** Time denotes average completion time per input in seconds. Confidence is self-reported on a 1–5 scale.

C.2 Human Reconstruction Baseline

To calibrate the difficulty of OrderProbe, we conduct a human reconstruction baseline on a sampled subset of the benchmark. Unlike the human audit in Appendix A.8, which evaluates the alignment between automated metrics and expert judgments of model explanations, this experiment directly measures whether humans can recover the canonical expression from the same scrambled inputs given to LLMs.

For each script setting, we sample a subset of canonical expressions and multiple non-identity permutations per expression. Participants are native speakers or highly proficient users of the corresponding language. Given a scrambled four-unit input, participants are asked to recover the canonical expression and optionally provide a short meaning explanation. No dictionary lookup or external search is allowed. We report exact recovery rate, average completion time, and self-reported confidence.

Human recovery is computed using the same exact-match criterion as model recovery:

$$\text{Human Recovery} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{x}_i^{human} = x_i). \tag{21}$$

This experiment provides an empirical reference point for interpreting model performance under structural perturbation.

Model	Setting	Recovery (%)	Parse Fail. (%)
DeepSeek-V3.2	+CoT	62.24 [60.71, 63.79]	0.9
Qwen-3-VLThink	+CoT	48.38 [46.72, 50.03]	1.4
Gemini-2.5-Flash	+CoT	41.56 [39.91, 43.18]	1.8
GPT-4o	+CoT	37.69 [36.08, 39.31]	2.2
Qwen-3-14B	Zero-shot	31.06 [29.58, 32.55]	1.1
Qwen-3-VLThink	Zero-shot	30.93 [29.47, 32.41]	1.3
Llama-3-8B	+CoT	24.43 [23.04, 25.86]	4.7
Qwen-3-8B	+CoT	12.59 [11.48, 13.73]	5.6

Table 11: **Uncertainty estimates and parsing failure rates.** Recovery is reported with bootstrap 95% confidence intervals computed by resampling canonical expressions rather than individual permutations. Parse Fail. denotes the percentage of outputs from which no valid recovered expression can be extracted.

Frequency Bin	Recovery (%)
High	30.576
Mid	19.794
Low	10.231

Table 12: **Pilot Frequency-Stratified Recovery (Zero-shot).** Exact recovery rates (%) across high-, mid-, and low-frequency bins.

C.3 Uncertainty Estimation and Parsing Reliability

We estimate uncertainty for the main recovery results using non-parametric bootstrap resampling. Because the 23 perturbed variants generated from the same canonical expression are not independent, we resample at the canonical-expression level rather than at the individual-permutation level. For each bootstrap sample, all associated permutations of the sampled expressions are included, and recovery rate is recomputed. We report 95% confidence intervals over $B = 1000$ bootstrap samples.

We also record parsing reliability. A model output is considered parseable if the recovered expression can be extracted from the predefined output field or, in the baseline setting, from the first valid four-unit candidate matching the script constraint. Unparseable outputs are counted as incorrect for recovery. We report parser failure rate separately to distinguish genuine reconstruction failures from output-format failures.

For closed-source API models, we record the evaluation date, model identifier, and available deployment version. For open-source models, we record checkpoint name, inference framework, decoding parameters, and tokenizer version.

Freq. Bin	Full Signal	Reduced Signal	Minimal Signal
High	61.25	19.74	8.32
Mid	39.74	10.62	6.83
Low	18.89	7.26	3.85

Table 13: **Frequency-stratified reconstruction accuracy under input-signal ablation.** Exact recovery rates (%) are reported across high-, mid-, and low-frequency bins as semantic support in the input and prompt is progressively reduced.

C.4 Frequency-Stratified Recovery Pilot

To quantify the potential influence of memorization, we conduct a frequency-stratified analysis on a uniformly sampled subset of 100 expressions from the benchmark. Each item is assigned to a frequency bin based on occurrence statistics from large-scale web-corpus frequency proxies.

To further examine how lexical familiarity interacts with available semantic cues, we conduct an input-signal ablation analysis. Instead of modifying the scoring pipeline, we progressively reduce the semantic support available in the input and prompt while keeping the reconstruction objective unchanged. The full-signal condition preserves the original scrambled four-unit input and standard reconstruction prompt. The reduced-signal condition removes auxiliary semantic hints and uses a minimal recovery-only prompt. The minimal-signal condition further suppresses explanation-oriented cues and requires only the canonical reconstruction output.

For each condition, we measure exact recovery across high-, mid-, and low-frequency bins. This design separates the effect of lexical familiarity from artifacts of automated semantic scoring.

For each disruption level, we measure exact reconstruction accuracy across high-, mid-, and low-frequency bins. Table 13 reports the resulting frequency-stratified performance.

C.5 Expanded Frequency-Stratified Analysis

We further stratify expressions into high-, mid-, and low-frequency bins within each script setting. Frequency is estimated from large-scale web-corpus occurrence statistics or dictionary frequency proxies when direct corpus frequency is unavailable. Bins are constructed within each language to avoid cross-lingual frequency-scale mismatch.

We analyze whether frequency predicts exact recovery while controlling for language and syntactic category. Specifically, we fit a logistic regression

Language	High Freq.	Mid Freq.	Low Freq.
ZH-CN	36.8	25.4	14.7
ZH-TW	31.2	20.7	11.6
JA	29.5	19.8	10.9
KO	10.8	5.6	2.4
Average	30.2	19.5	9.9

Table 14: **Frequency-stratified recovery by script setting.** Exact recovery rates (%) are reported across high-, mid-, and low-frequency bins constructed within each language.

Predictor	Coef.	Std. Err.	p-value
log Frequency	0.42	0.06	$< .001$
ZH-TW vs. ZH-CN	-0.31	0.08	$< .001$
JA vs. ZH-CN	-0.37	0.09	$< .001$
KO vs. ZH-CN	-1.84	0.12	$< .001$
Parallel structure	0.28	0.07	$< .001$
Verb–Object structure	-0.19	0.06	.002

Table 15: **Logistic regression analysis of exact recovery.** The dependent variable is whether the canonical expression is exactly recovered. Predictors include expression frequency, script setting, and syntactic structure.

model:

$$\text{logit}\, P(y_i = 1) = \beta_0 + \beta_1 \log f_i + \beta_2 L_i + \beta_3 T_i + \epsilon_i, \tag{22}$$

where y_i indicates exact recovery, f_i is expression frequency, L_i denotes the script setting, and T_i denotes syntactic category. This analysis separates the effect of lexical familiarity from script typology and structural pattern.

C.6 Synthetic Control Construction

To further examine whether reconstruction performance primarily reflects memorized idiom retrieval, we construct a synthetic control set consisting of non-attested four-character expressions.

Synthetic items are generated by sampling characters from semantically related groups (e.g., natural phenomena or seasonal terms) and combining them into four-character sequences that do not correspond to entries in standard idiom dictionaries or commonly used lexicalized expressions. We manually verify that the resulting combinations are not conventional idioms.

For each synthetic expression, a canonical order is defined and the same permutation protocol used in OrderProbe is applied. Models are evaluated using the identical reconstruction prompts and diagnostic metrics as in the main benchmark.

Dataset	Exact (%)	Sem. Fid.
Idiomatic Expr.	30.4	0.81
Synthetic Expr.	16.8	0.63

Table 16: Reconstruction performance on idiomatic expressions and synthetic non-idiomatic expressions under the same permutation protocol.

Table 16 reports reconstruction accuracy and semantic fidelity on idiomatic items and synthetic expressions under this setting.

C.7 Frequency Under Semantic Signal Disruption

A potential concern is that exact recovery on four-character expressions may be influenced by memorization, since frequent idioms are more likely to appear in pretraining corpora. To quantify this effect, we conduct a pilot frequency-stratified analysis under the zero-shot setting using two representative models that span a wide capability range: GPT-4o and Qwen-3-8B. We stratify expressions into three frequency bins (high, mid, and low) and measure the recovery rate as exact canonical reconstruction. Table 12 reports bin-wise recovery averaged across the two models.

As shown in Table 12, we observe a clear monotonic frequency effect: recovery is highest on high-frequency expressions, decreases on mid-frequency expressions, and remains non-trivial on low-frequency expressions. This indicates that lexical frequency contributes to reconstruction performance, while meaningful recovery persists even in the low-frequency regime.

D Prompt Templates for Idiom Meaning Inference

All baseline runs use a fixed two-message prompting scheme consisting of a system instruction and a user query template. The system instruction constrains the output format to a single-line response in Chinese. The user template provides the target idiom and repeats the required output schema to reduce formatting deviations. Across different experimental conditions, only the prompt text is modified, while the model, decoding settings, batch processing logic, and output parsing rules remain unchanged. For multilingual settings, the same prompting structure is retained, with the prompt content adapted through direct translation or equivalent reformulation while preserving identical functional roles and formatting requirements.

To facilitate reproducibility, Table 17 presents the baseline prompt templates used for idiom meaning inference across languages. The prompts are shown in their actual formatted layout, including line breaks, to illustrate the structural constraints imposed on model outputs. No task-specific examples, demonstrations, or additional guidance are included beyond the required format specification.

All experiments are conducted using fixed inference settings. For the *baseline* condition, no task-specific examples or chain-of-thought instructions are provided beyond the one-line format constraint. For the *CoT* condition, we use a fixed two-line CoT template (Table 19) while keeping all decoding settings identical.

E Literal Interpretation and Idiomatic Retrieval

Figure 7 presents a representative failure case on the idiom *Qing Guo Qing Cheng*. Although the scrambled characters preserve strong lexical cues, some models still produce fluent but non-canonical literal interpretations based on character-level composition. In contrast, stronger models correctly retrieve the idiomatic metaphor, demonstrating that semantic plausibility and structural reconstruction can diverge sharply.

F Qualitative Analysis and Case Studies

In this section, we provide a qualitative visual analysis of model performance across different experimental settings. Figures 8 and 9 illustrate baseline (zero-shot) behaviors and highlight common failure modes such as hallucination and structural blindness. Figures 10 through 13 further compare Chain-of-Thought (CoT) and few-shot strategies for each language (ZH-CN, ZH-TW, JP, KO). These visualizations display the system prompts, user queries, and model outputs, and they illustrate the strict constraints imposed by our evaluation pipeline.

G Reference Performance on Canonical Inputs

To isolate the specific impact of structural perturbation, we conducted a control experiment using the canonical (unperturbed) forms of the expressions. This serves two critical purposes: establishing a performance upper bound for each model and validating the reliability of our automated diagnostic metrics.

Table 20 reports the diagnostic scores under this ideal condition. We observe that all evaluated models demonstrate high proficiency when the input structure is intact. For instance, semantic fidelity scores (S_{Acc}^{mean}) consistently range between 0.75 and 0.88 across different model tiers, and logical validity (S_{Logic}) remains high.

These results support two key conclusions regarding the main experiments:

1. **Knowledge Verification:** The high baseline performance confirms that the models possess the requisite lexical knowledge in their pre-training distribution. Therefore, the failures observed in the scrambled settings (Section 4.2) cannot be attributed to a lack of semantic understanding, but rather to a specific deficit in structural reconstruction.

2. **Metric Validation:** The high scores under canonical conditions validate our evaluation pipeline. They indicate that our automated metrics accurately reward correct semantic interpretations, ruling out the possibility that the low scores in the main experiments are artifacts of an overly punitive scoring mechanism.

The stark contrast between this high-performance baseline and the significant degradation observed under permutation (where recovery rates frequently drop below 35%) provides compelling evidence for the gap between meaning-oriented generation and exact structural reconstruction discussed in the main text.

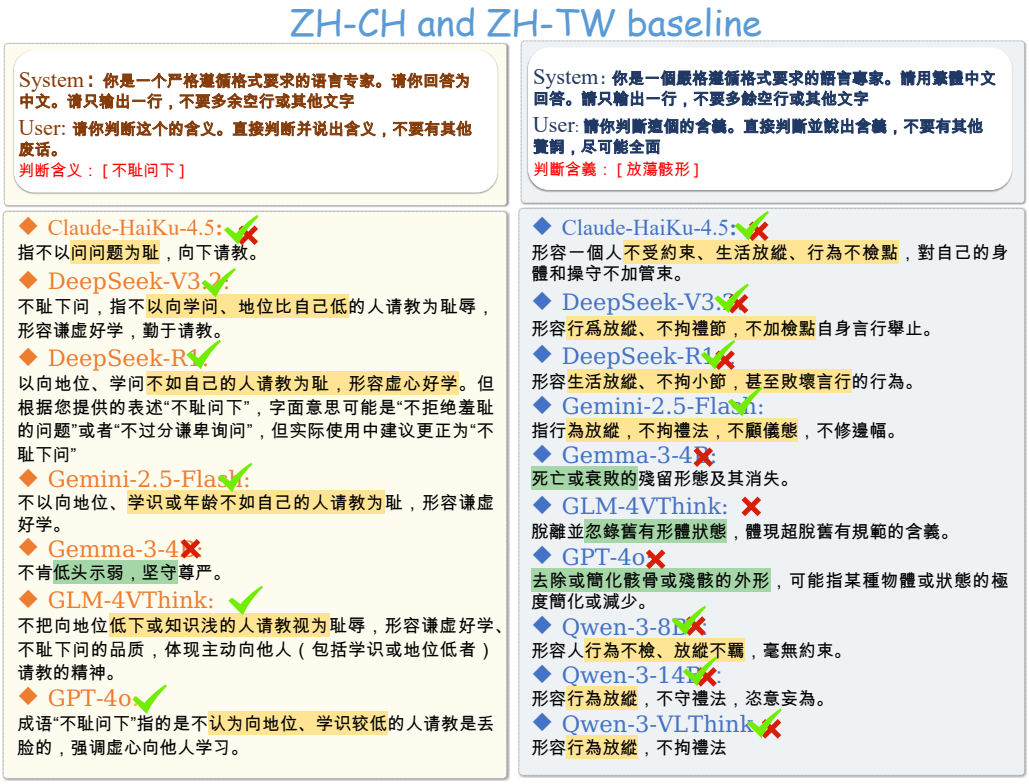

Figure 8: **Baseline performance on Chinese scripts illustrating Simplified Chinese on the left and Traditional Chinese on the right.** Even powerful models often fail to recover the canonical order.

Figure 9: **Baseline performance on Japanese and Korean scripts showing Korean Hangul on the left and Japanese Kanji on the right.** The structural scrambling effectively disrupts local semantic anchors.

Figure 10: **Comparison of prompting strategies on Simplified Chinese.** Chain of Thought prompting elicits better reasoning but still suffers from hallucination during complex reordering.

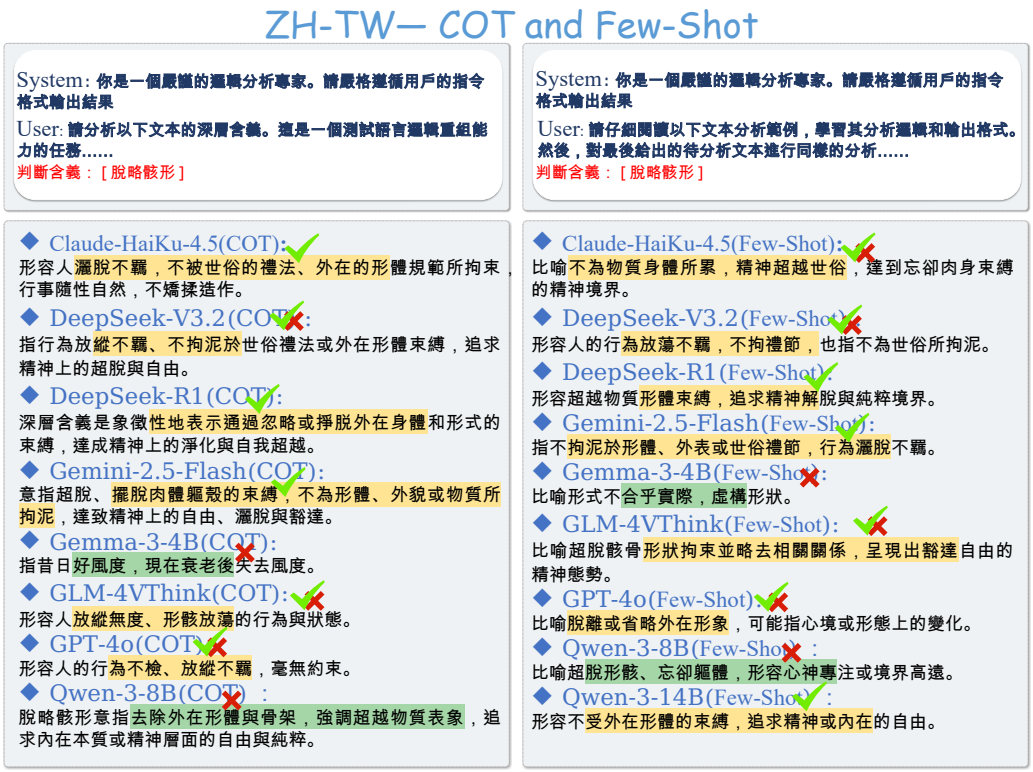

Figure 11: **Comparison of prompting strategies on Traditional Chinese.** The results highlight significant performance gaps across different inference methods.

Figure 12: **Comparison of prompting strategies on Japanese.** This detailed analysis highlights the syntax recovery challenges in mixed script inputs.

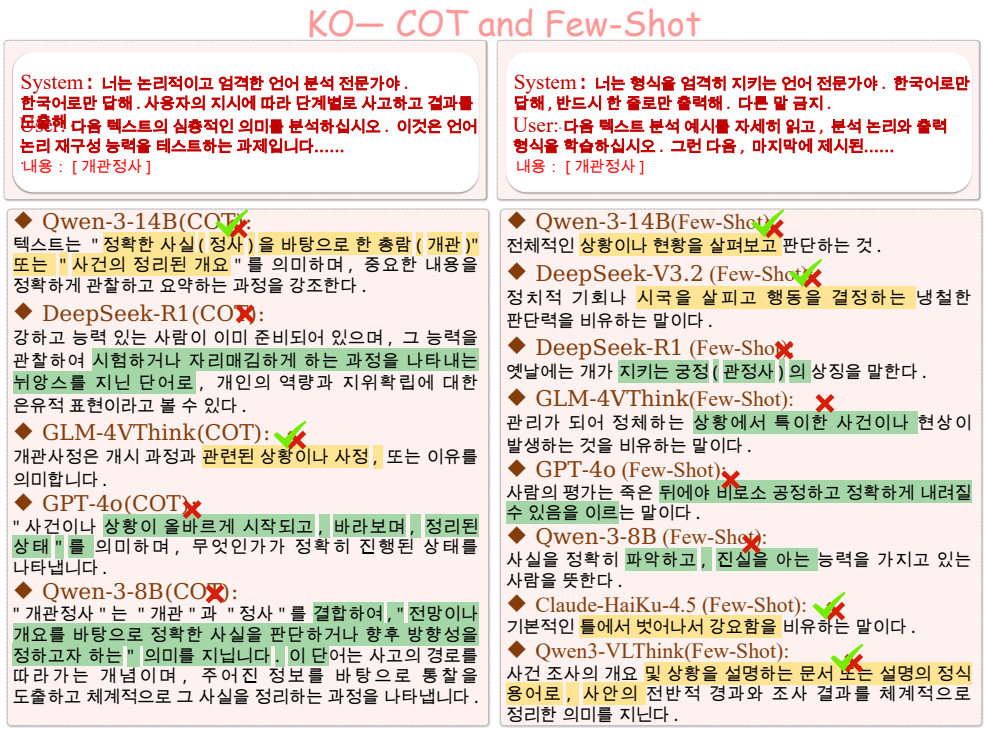

Figure 13: **Comparison of prompting strategies on Korean.** We observe distinct behaviors between standard prompting and reasoning enhanced generation.

Language	Type	Baseline Prompt Template (Visualized Layout)
ZH-CN	System	你是一个严格遵循格式要求的语言专家。请你回答为中文。请只输出一行，不要多余空行或其他文字： 判断含义：**[解释]**
	User	请你判断这个的含义。直接判断并说出含义，不要有其他废话，尽可能全面： 判断含义：**[解释]** 成语：{idiom}
ZH-TW	System	你是一個嚴格遵循格式要求的語言專家。請用繁體中文回答。請只輸出一行，不要多餘空行或其他文字： 判斷含義：**[解釋]**
	User	請你判斷這個的含義。直接判斷並說出含義，不要有其他贅詞，可能全面： 判斷含義：**[解釋]** 成語：{idiom}
KO	System	너는 형식을 엄격히 지키는 언어 전문가다. 한국어로만 답하라. 반드시 한 줄로만 출력하고 다른 문장은 포함하지 마라: **의미: [설명]**
	User	다음에 제시된 표현의 의미를 한 줄로 설명하라. 군말은 포함하지 말고 가능한 한 완전하게 설명하라: **의미: [설명]** 내용:{idiom}
JA	System	あなたは形式を厳守する言語専門家です。日本語で回答してください。必ず一行のみを出力し余分な文言や空行は含めないでください： 意味：**[説明]**
	User	以下に示す表現の意味を一行で説明してください。余分な文言は禁止し可能な限り完全に述べてください： 意味：**[説明]** 内容：{idiom}

Table 17: Visualized baseline prompt templates used for idiom meaning inference across four languages. Prompts are shown in their formatted layout to illustrate the structural constraints imposed on model outputs.

Language	Type	Few Shot Prompt Template Visualized Layout
ZH-CN	System	你是一个严谨的逻辑分析专家。请严格遵循用户的指令格式输出结果。禁止输出任何开场白或结束语。
	System	start deep thinking, please always use the thinking mode
	User	请仔细阅读以下文本分析示例，学习其分析逻辑和输出格式。然后，对最后给出的待分析文本进行同样的分析。 示例1 输入文本：惊人一鸣 Thinking Process：分析字面，惊人一鸣在语义上不通顺。观察字词组合，发现一鸣通常作为主语或状语，惊人作为结果。调整语序为一鸣惊人，符合一声鸣叫让人震惊的逻辑。 Final Meaning：比喻平时没有突出的表现，一下子做出惊人的成绩。 示例2 输入文本：添足画蛇 Thinking Process：分析字面，添足意味着增加脚，画蛇是动作主体。逻辑上先有画蛇，后有添足的多余行为。重组为画蛇添足。 Final Meaning：比喻做了多余的事，非但无益，反而不合适。 示例3 输入文本：求剑刻舟 Thinking Process：分析字面，求剑是目的，刻舟是手段。按照因果和动作发生的时间顺序，应先在船上刻记号，再去寻找剑。重组为刻舟求剑。 Final Meaning：比喻拘泥不知变通，不懂得根据实际情况处理问题。 待分析文本 输入文本：{idiom} 请严格模仿上述示例的逻辑和格式输出，必须严格两行。
ZH-TW	System	你是一個嚴謹的邏輯分析專家。請嚴格遵循用戶的指令格式輸出結果。禁止輸出任何開場白或結束語。
	System	start deep thinking, please always use the thinking mode
	User	請仔細閱讀以下文本分析範例，學習其分析邏輯和輸出格式。然後，對最後給出的待分析文本進行同樣的分析。 範例1 輸入文本：驚人一鳴 Thinking Process：分析字面，驚人一鳴在語義上不通順。觀察字詞組合，發現一鳴通常作為主語或狀語，驚人作為結果。調整語序為一鳴驚人，符合一聲鳴叫讓人震驚的邏輯。 Final Meaning：比喻平時沒有突出的表現，一下子做出驚人的成績。 範例2 輸入文本：添足畫蛇 Thinking Process：分析字面，添足意味著增加腳，畫蛇是動作主體。邏輯上先有畫蛇，後有添足的多餘行為。重組為畫蛇添足。 Final Meaning：比喻做了多餘的事，非但無益，反而不合適。 範例3 輸入文本：求劍刻舟 Thinking Process：分析字面，求劍是目的，刻舟是手段。按照因果和動作發生的時間順序，應先在船上刻記號，再去尋找劍。重組為刻舟求劍。 Final Meaning：比喻拘泥不知變通，不懂得根據實際情況處理問題。 待分析文本 輸入文本：{idiom} 請嚴格模仿上述範例的邏輯和格式輸出，必須嚴格兩行。
KO	System	너는 논리적이고 엄격한 언어 분석 전문가야. 한국어로만 답해. 사용자의 지시에 따라 단계별로 사고하고 결과를 도출해.
	System	start deep thinking, please always use the thinking mode
	User	다음 텍스트 분석 예시를 자세히 읽고 분석 논리와 출력 형식을 학습하십시오. 그런 다음 마지막에 제시된 분석 대상 텍스트에 대해 동일한 분석을 수행하십시오. 예시 1 입력 텍스트: 경인일명 (驚人一鳴) Thinking Process: 새가 한 번 울어 사람들을 놀라게 한다는 인과 관계가 자연스럽다. 따라서 어순을 일명경인으로 재구성한다. Final Meaning: 평소에는 조용하다가 갑자기 놀라운 성과를 내어 세상을 놀라게 함을 비유하는 말이다. 예시 2 입력 텍스트: 첨족화사 (添足蛇) Thinking Process: 뱀을 그리고 나서 발을 더하는 행위가 불필요하다. 어순을 화사첨족으로 재구성한다. Final Meaning: 쓸데없는 일을 덧붙여서 도리어 일을 그르침을 이르는 말이다. 예시 3 입력 텍스트: 구검각주 (求劍刻舟) Thinking Process: 칼을 찾기 위한 수단으로 배에 표시를 하므로 먼저 각주하고 나중에 구검한다. 어순을 각주구검으로 재구성한다. Final Meaning: 변화하는 상황을 모르고 낡은 방식에 집착하는 어리석음을 비유한다. 분석 대상 텍스트 입력 텍스트: {idiom} 위 예시들의 논리와 형식을 엄격히 모방하여 출력하십시오. 반드시 두 줄만 출력하십시오.
JA	System	あなたは厳格な論理を持つ言語分析の専門家です。ユーザーの指示するフォーマットに厳密に従い、余計な挨拶や結びの言葉は出力しないでください。
	System	start deep thinking, please always use the thinking mode
	User	次のテキスト分析例を注意深く読み、分析の論理と出力形式を学習してください。その後、最後に示される分析対象テキストについて同様の分析を行ってください。 例 1 入力テキスト：驚人一鳴 Thinking Process: 驚人一鳴は不自然であり、一鳴が原因で驚人が結果となるため語順を一鳴驚人に調整する。 Final Meaning: 普段は目立たないが突然すばらしい成果を出して人々を驚かせることのたとえ。 例 2 入力テキスト：添足画蛇 Thinking Process: まず蛇を描き、その後で足を足す行為が余計である。語順を画蛇添足に再構成する。 Final Meaning: 余計なことをしてかえって不適切になることのたとえ。 例 3 入力テキスト：求剣刻舟 Thinking Process: 剣を探すために舟に印を刻すので先に刻舟し後に求剣する。語順を刻舟求剣に再構成する。 Final Meaning: 状況の変化を理解せず古い方法に固執する愚かさのたとえ。 分析対象テキスト 入力テキスト：{idiom} 上記の例の論理と形式を厳密に模倣して出力してください。必ず二行のみを出力してください。

Table 18: Visualized few shot prompt templates used for multilingual idiom analysis. Prompts are shown in formatted layout to illustrate in context reasoning patterns and the strict two line output constraint.

Language	Type	CoT Prompt Template Visualized Layout
ZH-CN	System	你是一个严谨的逻辑分析专家。请严格遵循用户的指令格式输出结果。禁止输出任何开场白或结束语。 start deep thinking, please always use the thinking mode
	User	请分析以下文本的深层含义。这是一个测试语言逻辑重组能力的任务。 为了确保准确性，请务必一步步思考，严格遵循以下分析路径。 1 字义拆解分析文本中每个字的独立含义。 2 结构重组观察这些字之间可能存在的句法逻辑，判断是否存在语序错位。 3 语义推导基于重组后的逻辑，推导该文本的整体寓意。 请严格按照以下特定格式输出，必须严格两行。 Thinking Process: 在这里写下你的详细分步推理过程，包括对字序和结构的分析。 Final Meaning: 基于推理得出的最终含义解释，只给结论，不要再解释推理过程，尽可能全面。 输入文本：{idiom}
ZH-TW	System	你是一個嚴謹的邏輯分析專家。請嚴格遵循用戶的指令格式輸出結果。禁止輸出任何開場白或結束語。 start deep thinking, please always use the thinking mode
	User	請分析以下文本的深層含義。這是一個測試語言邏輯重組能力的任務。 為了確保準確性，請務必一步步思考，嚴格遵循以下分析路徑。 1 字義拆解分析文本中每個字的獨立含義。 2 結構重組觀察這些字之間可能存在的句法邏輯，判斷是否存在語序錯位。 3 語義推導基於重組後的邏輯，推導該文本的實際整體寓意。 請嚴格按照以下特定格式輸出，必須嚴格兩行。 Thinking Process: 在這裡寫下你的詳細分步推理過程，包括對字序和結構的分析。 Final Meaning: 基於推理得出的最終含義解釋，只給結論，不要再解釋推理過程，盡可能全面。 輸入文本：{idiom}
KO	System	너는 논리적이고 엄격한 언어 분석 전문가야. 한국어로만 답해. 사용자의 지시에 따라 단계별로 사고하고 결과를 도출해. start deep thinking, please always use the thinking mode
	User	다음 텍스트의 심층적인 의미를 분석하십시오. 이것은 언어 논리 재구성 능력을 테스트하는 과제입니다. 정확성을 보장하기 위해 반드시 단계별로 사고하고 다음 분석 경로를 엄격히 따르십시오. 1 글자 풀이 텍스트에 포함된 각 글자의 개별적인 의미를 분석합니다. 2 구조 재구성 글자들 간의 잠재적인 구문 논리를 관찰하여 어순이 뒤섞여 있는지 판단합니다. 3 의미 도출 재구성된 논리를 바탕으로 텍스트의 전체 의미를 추론합니다. 다음 형식에 맞춰 엄격히 출력하십시오. 반드시 두 줄이어야 하며 줄을 추가하거나 빼지 마십시오. Thinking Process: 여기에 글자 순서와 구조 분석을 포함한 상세한 단계별 추론 과정을 작성하십시오. Final Meaning: 추론을 통해 도출된 최종 의미 설명. 결론만 제시하고 추론 과정을 반복하지 마십시오. 최대한 포괄적으로 작성하십시오. 입력 텍스트: {idiom}
JA	System	あなたは厳格な論理を持つ言語分析の専門家です。ユーザーの指示するフォーマットに厳密に従い、余計な挨拶や結びの言葉は出力しないでください。 start deep thinking, please always use the thinking mode
	User	以下のテキストの深層的な意味を分析してください。これは言語論理の再構築能力をテストするタスクです。 正確性を確保するため必ずステップバイステップで思考し以下の分析パスを厳守してください。 1 文字の解読 テキストに含まれる各文字の個別の意味を分析します。 2 構造の再構築 文字間の構文論理を観察し語順が入れ替わっていないか判断します。 3 意味の導出 再構築された論理に基づいてテキスト全体の意味を推論します。 以下の形式に厳密に従って出力してください。必ず二行でなければなりません。 Thinking Process: ここに文字の順序と構造の分析を含む詳細な段階的推論プロセスを記述してください。 Final Meaning: 推論に基づいて導き出された最終的な意味の説明。結論のみを提示し推論過程を繰り返さないでください。可能な限り包括的に記述してください。 入力テキスト：{idiom}

Table 19: Visualized CoT prompt templates used for multilingual idiom analysis. The prompts enforce a fixed two line output format and instruct stepwise reasoning for semantic and structural reconstruction.

Model Family	S_{Acc}^{mean}	S_{Cons}	S_{Logic}	S_{Rob}	S_{Info}	Recovery (%)
Tier 1: High-Performance Reasoning Models						
Qwen-3-14B	0.862	0.857	0.873	0.904	0.841	**100**
Qwen-3-VLThink	0.884	0.853	0.886	0.913	0.832	**100**
Gemini-2.5-Flash	0.871	0.864	0.859	0.898	0.821	**100**
Tier 2: Mid-Range General Models						
GPT-4o	0.879	0.882	0.858	0.914	0.843	**100**
GLM-4VThink	0.833	0.842	0.845	0.883	0.801	**100**
DeepSeek-V3.2	0.874	0.868	0.893	0.909	0.847	**100**
Tier 3: Efficient & Baseline Models						
DeepSeek-R1	0.821	0.848	0.824	0.879	0.793	**100**
Claude-HaiKu-4.5	0.843	0.861	0.834	0.887	0.802	**100**
Gemma-3-4B	0.783	0.819	0.802	0.847	0.742	**100**
Llama-3-8B	0.758	0.832	0.791	0.853	0.723	**100**
Llama-2-7B	0.742	0.814	0.774	0.838	0.703	**100**
Qwen-3-8B	0.779	0.823	0.788	0.852	0.739	**100**

Table 20: **Reference Diagnostic Metrics under Unperturbed Inputs.** This appendix table reports diagnostic component scores obtained from a reference evaluation run using unperturbed inputs and the same inference configurations as the main experiments. Since the inputs are already in canonical order, exact reconstruction is trivially easier in this setting, and all evaluated models achieve perfect recovery on these inputs. These results are provided for contextual reference only and are not used in any comparative or statistical claims.