Phun-Bench: Evaluating LLMs on Phonological Understanding in Chinese

Xing Yue   Yongliang Shen   Weiming Lu*
Zhejiang University
{yue_xing, syl, luwm}@zju.edu.cn

Abstract

Language is a vehicle for thought, intricately tied to sounds, symbols, and meaning. However, most large language model (LLM) research focuses on meaning (semantics) and symbols (spelling) while largely overlooking sounds. Existing benchmarks on LLMs' phonological abilities are either solvable through rote memorization or intertwined with other abilities, making them inadequate to measure LLMs' genuine ability in *phonological understanding*. Here, we present Phun-Bench, a purpose-built Chinese benchmark with diverse tasks and settings across three dimensions (Homophony, Rhyme, and Phonetic Similarity), designed to systematically evaluate LLMs' phonological understanding. Our results show that while LLMs excel at recalling correct pronunciations, they generally struggle to leverage phonological knowledge in the flexible and intuitive way that human speakers do. Moreover, through detailed analyses, we propose a hypothesis regarding the underlying mechanism of LLMs' phonological understanding and "perception", highlighting an underexplored frontier for future research. [1]

1 Introduction

Large language models (LLMs), trained on vast corpora and refined through post-training, have shown impressive performance across domains. With chain-of-thought (CoT, Wei et al., 2022) prompting and reinforcement learning (RL), large reasoning models (LRMs, DeepSeek-AI et al., 2025; OpenAI, 2025) can now tackle tasks such as mathematics, coding, QA, and creative writing, often matching or surpassing human-level performance.

From a linguistic perspective, language can be viewed as the integration of three aspects: *sound* (phonology and phonetics), *symbol*[2] (orthography and morphology), and *meaning* (semantics). While the meaning aspect alone suffices for many tasks, there exist practically important scenarios, such as creative writing (Ormazabal et al., 2022), spelling error correction (Liang et al., 2023), and toxicity detection under homophonic cloaking (Lu et al., 2023; Xiao et al., 2024), where a joint understanding of both sound and meaning is essential. For instance, without an understanding of rhyme, we cannot expect a model to generate a well-formed poetic line.

However, LLMs are pre-trained on text corpora and intrinsically lack audio-modal capabilities in input, reasoning, and output. A natural question arises: can LLMs acquire an understanding of sound as an emergent ability? We find that few prior studies have directly evaluated LLMs' phonological understanding or investigated its underlying mechanism. Existing work has typically focused on specific downstream tasks that rely on phonological ability, such as creative writing (e.g., poetry (Ormazabal et al., 2022), lyrics (Yuan et al., 2024), and puns (Chen et al., 2024a; Su et al., 2025)), humor understanding (He et al., 2025), toxicity detection (Lu et al., 2023; Mou et al., 2020), and Chinese spelling correction (Li et al., 2022; Liang et al., 2023). Yet, in such tasks it is difficult to disentangle the contribution of phonological ability from other capacities, such as semantic knowledge. Some studies instead assess LLMs' basic phonological knowledge—for instance, grapheme-to-phoneme (G2P, Cheng et al., 2024) and phoneme-to-grapheme (P2G, Lauc, 2025) conversion, phonological-rule QA (Shin et al., 2025), or rhyme word generation (Suvarna et al., 2024). However, these tasks can often be solved through rote memorization of phono-

*Corresponding author.

[1] We release our dataset and code at https://github.com/xing-stellus-yue/Phun-Bench.

[2] LLMs also struggle with basic symbol-related tasks such as letter counting (Fu et al., 2024; Xu and Ma, 2025) and Chinese character decomposition (Wu et al., 2025), which is beyond our scope.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 22723–22768
July 2-7, 2026 ©2026 Association for Computational Linguistics

logical knowledge (e.g., via online dictionaries) rather than reflecting genuine **phonological understanding**. For example, an LLM may accurately recall word pronunciations, yet still struggle to judge whether two words sound similar or rhyme, an ability that human speakers acquire intuitively.

We first verify LLMs' extensive Chinese phonological knowledge with a character-to-pinyin test. We then introduce Phun-Bench, a Chinese benchmark purpose-built to evaluate LLMs' phonological understanding across three dimensions (Homophony, Rhyme, and Phonetic Similarity), comprising four tasks: (1) Homophone Recall, (2) Contextual Homophone Recognition, (3) Rhyming Sentence Generation, and (4) Similarity Comparison. Evaluations on LLMs show that while they excel at recalling correct pronunciations, they generally underperform on tasks requiring phonological understanding. Through detailed analyses, we further reveal the challenges LLMs face and the underlying mechanism of their phonological understanding, namely, reasoning via the sound layer. Finally, we evaluate audio models on Phun-Bench, highlighting a promising multimodal direction for advancing phonological tasks.

To summarize, our main contributions are:

- (§ 3) We introduce the concept of *phonological understanding*, a missing piece in LLM research, and outline the criteria for its evaluation.
- (§ 4) Based on these criteria, we propose Phun-Bench, a purpose-built Chinese benchmark with diverse tasks and settings across three dimensions (Homophony, Rhyme, and Phonetic Similarity), and apply it to assess LLMs' phonological understanding.
- (§ 5) Through detailed analyses and additional experiments, we reveal the challenges LLMs face and the underlying mechanism of their phonological understanding, offering insights for related fields and future research.

2 Related Work

LLMs' Phonological Ability Early work on phonology in NLP primarily focuses on grapheme-to-phoneme (G2P) conversion (Cheng et al., 2024), a key component in text-to-speech (TTS) and automatic speech recognition (ASR). Notable datasets include CMUDict (CMU), g2pM (Park and Lee, 2020), and the SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme (Gorman et al., 2020). With the emergence of LLMs, some studies benchmark them on elementary language tasks, including morphology, phonology/phonetics and grammar. For example, LMentry (Efrat et al., 2023) features two phonological tasks (word rhyme judgment and homophone judgment) among other non-phonological tasks, and KoBALT (Shin et al., 2025) and KoGEM (Kim et al., 2025) include Korean phonological-rule tasks as part of broader benchmarks. To our knowledge, PhonologyBench (Suvarna et al., 2024) is the only benchmark purpose-built to evaluate LLMs' phonological ability in English, comprising three tasks: G2P conversion, syllable counting, and rhyme word generation. These tasks actually test memorization of phonological knowledge. In contrast, Phun-Bench focuses on tasks requiring genuine *phonological understanding*, and we further analyze LLMs' underlying mechanism of it.

Prior research often focuses on downstream tasks that incidentally require phonological ability, such as *pun recognition*, *rhyme-related generation*. A comprehensive review of these works is provided in Appendix B. In particular, HomoP-CN (Ma et al., 2025) evaluates LLMs' ability to recover Chinese Internet homophones and distinguishes between reasoning and memorization through perturbation. While our work shares some similarities in isolation design and methodology, we instead probe LLMs' intrinsic phonological ability directly and systematically.

3 Phonological Understanding

3.1 Definition

We propose *phonological understanding*, defined as <u>the ability to flexibly apply the phonological and phonetic knowledge inherent in the language in novel situations, beyond simple recall</u>, which serves as a prerequisite for more complex reasoning tasks, such as poetry generation and perturbed toxicity detection. This definition aligns with the distinction between remembering and understanding in Bloom's Taxonomy (Bloom et al., 1956) and between declarative and procedural knowledge proposed by ACT-R (Anderson et al., 1997), and follows a functional view that understanding can be inferred from task performance (Goldstein and Stanovsky, 2024). For instance, native speakers can quickly grasp a homophonic pun, whereas even advanced second-language learners often struggle.

Question Pattern Design

Example 1: What's the pronunciation of the idiom 人杰地灵? Represent it in pinyin.

(solvable by rote memorization)

Improved: Which idiom is homophone to 仁截帝聆?

Example 2: Generate a rhymed poetry on spring.

(involving multiple abilities beyond phonology)

Improved: Generate a sentence that shares an end-rhyme with 我爱人工智能.

(a) We refine the task design from existing benchmarks.

Char.	Init.	Final	Tone	ST	L_{sim}
牛	n	iou	2	T_0	0
狗	g	ou	3	T_1	1
小	x	iao	3	T_2	1
书	sh	u	1	T_3	1
告	g	ao	4	T_4	2
歌	g	e	1	T_5	2
猫	m	ao	1	T_6	2
糕	g	ao	1	T_7	3

(b) Similarity Types (STs).

Figure 1: (a) Our question pattern design compared to those in prior work. (b) Each pinyin is decomposed into *initial*, *final*, and *tone*. We compare these components between two characters to determine the ST. For illustration, each character's ST is shown relative to 高 (*gao1*), with green for match and red for mismatch.

3.2 How to Evaluate Phonological Understanding?

While some existing work has investigated LLMs' phonological ability, they often fall short in evaluating LLMs' *phonological understanding*, as illustrated by the examples in Figure 1a.

- Some benchmarks target phonological evaluation, but can be solved through rote memorization of phonological knowledge, such as pronunciations or phonological rules.
- Other benchmarks focus on specific downstream tasks, such as spelling correction (Li et al., 2022; Liang et al., 2023), or poetry and lyrics generation. However, these tasks are not phonology-only in nature and typically rely heavily on semantic ability.

So, we propose that evaluating LLMs' genuine phonological understanding should adhere to two criteria:

1. **Memorization-Proof:** Minimize the possibility that LLMs can succeed merely through memorization.
2. **Solve with Sounds, not with Meaning:** Exclude the contribution to results from other abilities, such as semantic understanding.

Language Selection Additionally, for languages with alphabetic *writing systems* (e.g., English, Greek), the spelling (and thus the token) can itself be phonologically informative (as the results hint in RQ 3 (§ 5.4.2)), making logographic languages like Chinese preferable to reduce such shortcuts[3]. Moreover, Chinese offers a large pool of homophonic characters, each corresponding to exactly one syllable, which further benefits our benchmark design and data curation.

3.3 Char2pinyin: A Verification Experiment

Since no existing work evaluated LLMs on Chinese phonological skills, we begin by verifying their *phonological knowledge*, through the elementary task of **character-to-pinyin**[4] **(char2pinyin)** conversion, a foundation for phonological understanding. Specifically, we simply prompt LLMs to spell out the pinyin of given characters c_{query}.

We curate characters from the *List of Commonly-Used Standard Chinese Characters* (Ministry of Education and State Language Commission, 2013) and retain only monophonic characters. This yields 5,345 unique characters as c_{query}. Evaluation on both LLMs and humans show that LLMs generally outperform humans, exhibiting their extensive phonological knowledge. We present the detailed results in Appendix C.1.

4 Phun-Bench

4.1 Benchmark Design

We design Phun-Bench following the two Criteria (§ 3.2). Then, for non-audio LLMs, *how can we measure their phonological understanding?* Drawing inspiration from the distributional hypothesis

[3] Although radicals in Chinese characters can hint at pronunciation, LLMs often fail to decompose characters correctly (Xu and Ma, 2025), so we see little risk of "leakage."

[4] Pinyin is the official romanization system adopted in China to represent the pronunciations of characters and words. For an introduction to pinyin, see Appendix A.

popularized by J. R. Firth (Firch, 1957), who proposed that *a word is characterized by the company it keeps*, we argue that phonological understanding can be indicated by an understanding of the phonological and phonetic *relationships* between word pronunciations, such as phonetic similarity and rhymes. For example, humans can intuitively judge if two words sound alike and can easily construct a sentence that rhymes with another.

By analyzing the syllabic structure of Chinese, we categorize eight **similarity types (STs)** between two characters, as presented in Figure 1b, where T_7 denotes "identical" and T_0 denotes "totally different". We use L_{sim} to indicate the similarity level. Based on this categorization, we design Phun-Bench across three dimensions:

- (§ 4.2) **Homophony-Based Understanding:** the ability to recall commonly-used words from their homophones (T_7), either in isolation or within context.
- (§ 4.3) **Rhyme-Based Understanding:** the ability to generate a sentence that shares a single-rhyme or double-rhyme ($T_{2,\,4,\,6,\,7}$) with a given reference sentence.
- (§ 4.4) **Phonetic-Similarity-Based Understanding:** the ability to compare the phonetic similarity between given words ($T_{1\text{–}7}$).

These three dimensions are representative rather than exhaustive. We prioritize homophony, rhyme, and phonetic similarity because they recur in downstream tasks (spell checking creative generation, moderation) and map to established constructs: homophony as the basic equivalence relation, phonetic similarity as a graded relation studied in psychoacoustic and embedding-based work (Fang et al., 2020), and rhyme as a core literary device. Task samples of each dimension are presented in Figure 2, and the statistics are in Appendix C.3.

4.2 Dimension 1: Homophony-Based Understanding

As building connections between words with homophonic relationship is essential to phonological understanding, we evaluate LLMs on recalling words from their homophones, via two tasks.

4.2.1 Homophone Recall

Task Formulation Here, we use *chengyu*[5] as the target word w_{target}. Specifically, we provide LLMs with an auto-generated homophone w_{ref} and prompt them to recover the original chengyu. Since w_{ref} is probabilistically unlikely to appear in the pre-training corpus and is semantically nonsensical, this approach makes the dataset memorization-proof and eliminates the influence of semantic ability. We use **exact match (EM)** score to evaluate performance, which indicates if $\hat{w}_{\text{target}} = w_{\text{target}}$.

Dataset Construction We obtain chengyu from `chinese-idiom-db`[6] and retain only four-character entries. The chengyu are then sorted by frequency in the `news2016zh`[7] corpus in decreasing order, and the 2,857 most commonly-used are selected as w_{target} ($freq \geq 10$). To generate w_{ref}, we algorithmically perturb each w_{target} with homophonic commonly-used characters when available (see Algorithm 1). This results in 1,098 $(w_{\text{ref}},\ w_{\text{target}})$ tuples.

4.2.2 Contextual Homophone Recognition

Task Formulation We adapt the homophonic pun detection task for our evaluation. Specifically, LLMs are provided with a pun sentence s and are instructed to (1) detect the *pun word* w_{pun} within s, and (2) recover the corresponding *alternative word* w_{target}. This two-step design enables us to disentangle the contribution of semantic knowledge from phonological understanding. Following Kim et al. (2024), we evaluate using the **cover exact match (CEM)** score for two metrics: **detection rate (DR)**, i.e., if $\hat{w}_{\text{pun}}$ is covered by w_{pun}, and the **recovery rate (RR)**, i.e., if $\hat{w}_{\text{target}}$ is covered by w_{target}.

Dataset Construction We adapt pun sentences collected by Rohn (2024) as s, and then filter out those where w_{pun} is shorter than three characters or require niche-domain knowledge (e.g., pop culture). Using the `jieba`[8] and `pypinyin`[9] toolkits, we convert w_{pun} and w_{target} into pinyin p_{pun} and p_{target}, and manually correct any errors. Only strictly homophonic puns, where $p_{\text{pun}} = p_{\text{target}}$, are retained.

[5] A type of traditional Chinese idiomatic expressions, typically composed of four characters. For an introduction to chengyu, see Appendix A.

[6] https://github.com/by-syk/chinese-idiom-db

[7] https://github.com/brightmart/nlp_chinese_corpus

[8] https://github.com/fxsjy/jieba

[9] https://github.com/mozillazg/python-pinyin

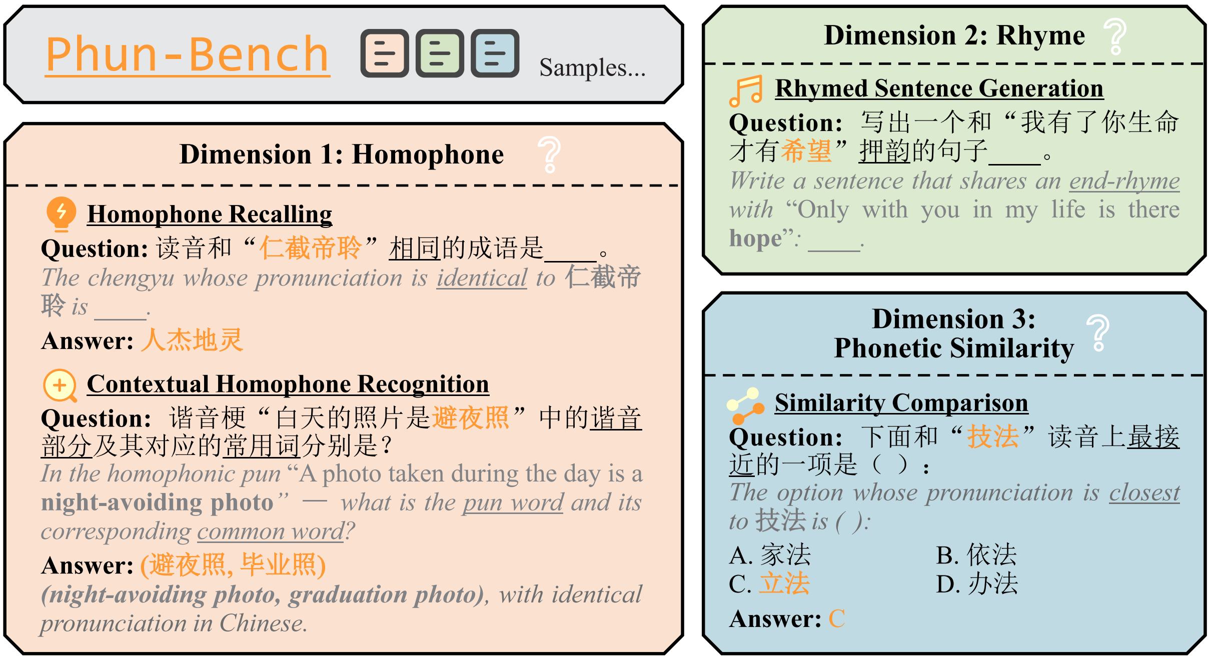

Figure 2: Data samples from Phun-Bench.

This results in 1,684 $(s,\ w_{\text{pun}},\ w_{\text{target}})$ tuples.

4.3 Dimension 2: Rhyme-Based Understanding

We present the **Rhyming Sentence Generation (RSG)** task to assess LLMs' rhyme understanding.

Task Formulation Generating rhyming sentences requires an understanding of the language's phonological system and the relationships between rhyming words. Specifically, we condition LLMs on a reference sentence s_{ref} and a rhyme rule $\mathcal{R}$, and instruct them to compose a sentence s_{gen} that rhymes with s_{ref} (i.e., belong to the same **rhyme group (RG)** defined in $\mathcal{R}$). To prevent shortcuts via memorizing rhyming words or lyrics, we introduce both a standard **single-rhyme (1-rhyme)** setting and a more challenging **double-rhyme (2-rhyme)** setting, in which the last *two* syllables/characters of s_{gen} must rhyme with those of s_{ref}.

Dataset Construction We manually define the rhyme rule $\mathcal{R}$, dividing all syllables into 18 RGs ($\mathcal{R} \equiv \{G_i \mid i = 1, 2, ..., 18\}$). We then adapt the lyrics from `ChineseLyrics`[10] as s_{ref}, filtering out: (1) non-lyric notations (e.g., "chorus" or "songwriter"), (2) sentences containing non-Chinese characters or alphanumerics, and (3) sentences shorter than six characters. Using `jieba` and `pypinyin`, we convert qualified s_{ref} to pinyin sequences and obtain their corresponding RGs as G_{ref}. We first randomly sample about 3,000 sentences, then deduplicate and manually filter them to obtain 1,000 $(s_{\text{ref}},\ G_{\text{ref}})$ tuples.

[10] https://github.com/dengxiuqi/ChineseLyrics

PT	Ans.	Dist.	Example w_{ref}	Example w_{cand}
$Pert_1$	$w_{7;\,6}$	$w_{7;\,1\text{–}3}$	对立	费力 案例 威力 霹雳
$Pert_2$	$w_{7;\,5}$	$w_{7;\,1\text{–}3}$	费用	服用 运用 没用 备用
$Pert_3$	$w_{7;\,4}$	$w_{7;\,1\text{–}3}$	纪要	将要 需要 机要 概要
$Pert_4$	$w_{4\text{–}6;\,4\text{–}6}$	$w_{1\text{–}3;\,1\text{–}3}$	私藏	脂肪 剥离 十足 安装

Table 1: Perturbation types (PTs) employed in the SC task. green denotes answer, and red denotes distractors.

Metrics Each generate sentence s_{gen} is converted to pinyin to obtain its RG G_{gen}. We then compute the EM score between G_{ref} and G_{gen}. Notably, we observe that some models tend to "plagiarize" the ending characters from the reference. To prevent this, we explicitly proscribe such behavior in the instruction and disqualify any such generations.

4.4 Dimension 3: Phonetic-Similarity-Based Understanding

Building on the proposed STs (§ 4.1), we introduce the **Similarity Comparison (SC)** task and generate word perturbations (Ma et al., 2025) algorithmically to evaluate LLMs' understanding of phonetic similarity.

Task Formulation We condition LLMs on a reference word w_{ref} and four candidate words (denoted w_{cand_i}, $i = 1, 2, 3, 4$), and instruct them to

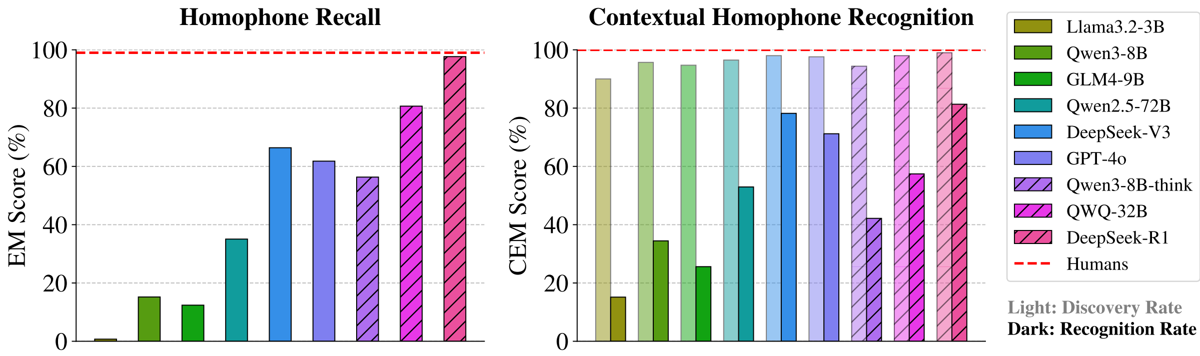

Figure 3: Results of the two homophony-based understanding tasks—HR and CHR. In the right panel, light bars indicate discovery rate (DR) and dark bars indicate recognition rate (RR).

select the word (denoted w_{close}) that is *most* phonetically similar to w_{ref}. Both w_{ref} and each w_{cand_i} are provided either in character form or pinyin form. We use EM score to evaluate performance.

Dataset Construction We begin with two-character words from *List of Frequently Used Words in Modern Chinese* (State Language Commission, 2008). These words are sorted by frequency in the `news2016zh` corpus and the 13,741 most commonly-used words ($freq \geq 100$) are retained. We then apply the **perturbation types (PTs)** listed in Table 1 to generate candidate words w_{cand_i} (see Algorithm 2). For example, in the $Pert_1$ setting, we identify two types of words: $w_{7;\,6}$, which contains one T_7 character and one T_6 character, and $w_{7;\,1\text{–}3}$, which contains one T_7 character and one $T_{1\text{–}3}$ character, aligned with the characters in w_{ref}. We generate one $w_{7;\,6}$ as the correct answer (denoted w_{close}) and three $w_{7;\,1\text{–}3}$ words as distractors (denoted w_{far}) for each w_{ref} (when available), which are then shuffled to form the candidate set w_{cand_i}.

5 Experiment

5.1 Experiment Settings

Evaluated Models We evaluate the proposed *phonological understanding* (§ 3 and 4) on eight models with varying sizes and capabilities. `Qwen3-8B` is evaluated in both modes.

- **Non-Thinking Models:** Llama3.2-3B-Instruct (Meta AI, 2024), Qwen3-8B (Yang et al., 2025), GLM4-9B-Chat (Zeng et al., 2024), Qwen2.5-72B-Instruct-AWQ (Yang et al., 2024), DeepSeek-V3 (DeepSeek-AI et al., 2024) and GPT-4o (Hurst et al., 2024).
- **Thinking Models:** Qwen3-8B, QwQ-32B-AWQ (Qwen Team, 2025) and DeepSeek-R1 (DeepSeek-AI et al., 2025).

For implementation details, see Appendix D.1. Prompts can be found in Appendix I. Due to budget constraints, we run 300 samples per setting for thinking models, 500 for `DeepSeek-V3` and `GPT-4o`, and the full set for the others.

Human Evaluation For comparison, we invited three native Chinese undergraduate students to assess `Phun-Bench`. We randomly sampled 50–100 examples per setting and evenly distributed them among the participants for evaluation. Further details are in Appendix D.2.

5.2 Dimension 1: Homophony-Based Understanding

5.2.1 Results

The results in Figure 3 show that while humans achieve nearly 100% accuracy, LLMs demonstrate subpar performance in recovering words from their homophones, suggesting that LLMs do not associate homophonic words as naturally as humans. Notably, in the CHR setting, all LLMs achieve a high DR (>90%) but a substantially lower RR. This indicates that, although LLMs are proficient at detecting pun words using semantic knowledge, they struggle to recover the corresponding alternatives based on phonological understanding.

5.2.2 RQ 1: How Do LLMs Recover the Homophonic Target Word?

We hypothesize that *LLMs perform phonological understanding primarily via pronunciation*, or the *sound layer*, as illustrated in Figure 4. This hypothesis is motivated by distributional asymmetry in pre-training corpora: co-occurrence between a word and its pronunciation is far more frequent than between a word and a perturbation-generated homophone, which we further avoid in our data design. To test this hypothesis, we decompose the word2word task into two steps: word2pinyin and pinyin2word. Specifically, we prompt models (1) to convert w_{ref} (from HR) or w_{pun} (from CHR) into their pinyin p (i.e., word2pinyin), and (2) to recover w_{target} from its pinyin p (i.e., pinyin2word). The results are shown in Table 2. For HR, we observe that:

$$P(pron \mid ref) \cdot P(target \mid pron) \geq P(target \mid ref) \tag{1}$$

And for CHR:

$$P(ref \mid s) \cdot P(pron \mid ref) \cdot P(target \mid pron) \geq P(target \mid s) \tag{2}$$

These results suggest that LLMs do not effectively associate homophonic words; instead, an indirect two-step conversion via the phonological (sound) layer appears to be easier, thereby supporting our hypothesis empirically. Reasoning traces from thinking models further corroborate this finding (Appendix E.1). A similar investigation on the Character-Level Homophone Recall (CLHR) setting is reported in Appendix E.2.

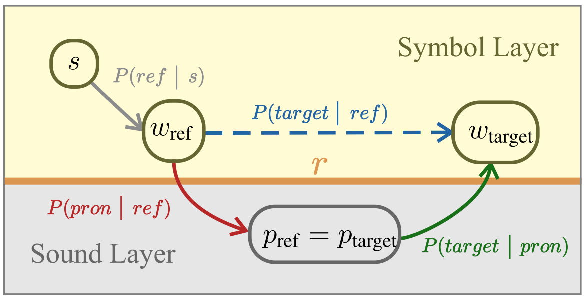

Figure 4: Illustration of the hypothesized mechanism of *phonological understanding via the sound layer*, as in the CHR task.

5.3 Dimension 2: Rhyme-Based Understanding

5.3.1 Results

The results are shown in Table 3. We observe that LLMs generally perform well in 1-rhyme generation but struggle with 2-rhyme generation, showing a clear performance drop. Reasoning appears necessary in this setting, as even humans require more time to compose sentences under complex rhyme constraints.

5.3.2 RQ 2: What Challenges Do LLMs Face in RSG?

Analysis of incorrect reasoning traces from `QwQ-32B` and `DeepSeek-R1` reveals two main error sources. The first is *indexing error*, where the model fails to correctly identify the last or next-to-last character in either the reference or the generated sentence. The second arises from *misgrouping* characters into incorrect RGs, reflecting a flawed understanding of the Chinese phonological system. For instance, `DeepSeek-R1` may rhyme 去 (*qù*, G_6) with 古 (*gǔ*, G_7) due to their seemingly identical final *u*[11]. In addition, we observe that `Qwen3-8B-think` is prone to meaningless repetitions of certain sentences or characters, leading to abnormally disastrous performance. Further analyses are provided in Appendix F.

5.4 Dimension 3: Phonetic-Similarity-Based Understanding

As mentioned in § 4.4, we use the Similarity Comparison (SC) task to measure LLMs' phonological understanding.

5.4.1 Results

The results of SC (in word-form setting) are shown in Figure 5, illustrating that non-thinking models perform poorly, while thinking models achieve some improvement, though still generally below human-level performance (except for the $Pert_3$ setting). Additionally, we provide results for the pinyin-form setting where LLMs show improved performance; detailed results and comparisons are provided in Appendix G.2.

5.4.2 RQ 3: How Do LLMs "Perceive" Phonetic Similarity?

While humans typically judge phonetic similarity based on intuition, LLMs appear to rely on reasoning to arrive at their answers, supporting the hypothesis proposed in RQ 1 (§ 5.2.2). With reasoning, `Qwen3-8B-think` can even outperform `DeepSeek-V3` and `GPT-4o`. By examining the reasoning traces from `DeepSeek-R1` and `Qwen3-8B-think` (for details, see Appendix G.1), we observe that they often first convert w_{ref} and each w_{cand_i} into pinyin, then decompose each syllable into a $(initial, final, tone)$ tuple, and finally compare them using explicit phonological

[11] The actual final of 去 is *ü*, with the diacritic conventionally omitted after the initial *q*.

Task	Setting	Model									Relationship
		L3.2-3B	Q3-8B	GLM4-9B	Q2.5-72B	DS-V3	GPT-4o	Q3-8B-think	QwQ-32B	DS-R1	
HR	$P(p \mid w_q)$	0.049	0.956	0.898	0.955	1.000	0.996	0.972	0.930	1.000	$r = \frac{P_1 P_2}{P_3} \geq 1$
	$P(w_t \mid p)$	0.419	0.826	0.712	0.954	0.996	0.946	0.897	0.962	1.000	
	$P(w_t \mid w_q)$	0.007	0.152	0.124	0.351	0.664	0.618	0.563	0.807	0.977	
	r	2.83	5.19	5.23	2.63	1.50	1.52	1.55	1.13	1.02	
CHR	$P(w_q \mid s)$	0.900	0.957	0.947	0.965	0.980	0.976	0.944	0.980	0.990	$r = P_1 \frac{P_1 P_2}{P_3} \geq 1$
	$P(p \mid w_q)$	0.690	0.869	0.822	0.943	0.981	0.970	0.923	0.877	0.960	
	$P(w_t \mid p)$	0.310	0.591	0.531	0.796	0.925	0.868	0.668	0.824	0.977	
	$P(w_t \mid s)$	0.152	0.345	0.256	0.530	0.782	0.712	0.422	0.574	0.813	
	r	1.27	1.43	1.61	1.37	1.14	1.15	1.38	1.23	1.14	

Table 2: Accuracy for word2pinyin and pinyin2word conversion. Homophone Recall (HR) accuracy is shown as $P(w_t \mid w_q)$, while Contextual Homophone Recognition (CHR) discovery rate (DR) and recognition rate (RR) are represented as $P(w_q \mid s)$ and $P(w_t \mid s)$, respectively. L, Q, and DS denote `Llama`, `Qwen` and `DeepSeek`.

Setting	L3.2-3B	Q3-8B	GLM4-9B	Q.5-72B	DS-V3	GPT-4o	Q3-8B-think	QwQ-32B	DS-R1
1-Rhyme	0.111	0.399	0.160	0.555	0.816	0.827	0.126	0.830	0.970
2-Rhyme	0.014	0.051	0.009	0.053	0.083	0.072	0.022	0.437	0.848

Table 3: Performance of LLMs on the Rhyming Sentence Generation (RSG) task. L, Q and DS denote `Llama`, `Qwen` and `DeepSeek`, respectively.

knowledge. Although this "breaking-down" reasoning process differs substantially from the intuitive mode of human judgment, it can nevertheless be effective—particularly in $Pert_3$ settings, where w_{close} and w_{ref} differ only in tone, a distinction that can be easily captured at the token level. In contrast, changes in the initials (as in $Pert_1$) can "mislead" LLMs, leading to poor performance.

5.4.3 An Exploration: How Do Audio Models Perform?

If LLMs without audio-modal ability must rely on reasoning to compare phonetic similarity, a natural question is whether an audio-enabled LLM would perform better. To investigate, we randomly sample 50 instances from each perturbation type, convert them to audio files using TTS, and evaluate them with `Qwen-Omni-Turbo`[12] and `Qwen3-Omni-Flash` (Xu et al., 2025b). As shown in Figure 6, audio capability improves performance, and with reasoning, `Qwen3-Omni-Flash` (with ∼30B parameters) can even outperform `GPT-4o` (Xu et al., 2025b). Further analysis by perturbation type shows differing strengths between text-only and multimodal models (see Appendix G.3).

We also inspected thinking traces from `Qwen3-Omni-Flash`. Many traces first transcribe the audio (sometimes with similar-sounding errors) and then proceed with pinyin-based comparison, often code-switching into English. This suggests current multimodal models may still rely on text-centric reasoning, with audio embeddings underutilized for direct comparison; a deeper analysis on the mechanism of multimodal model's phonological understanding is left for future work.

[12]Reportedly based on `Qwen2.5-Omni` (Xu et al., 2025a).

5.5 Scaling of Phonological Understanding

To analyze how LLMs' phonological understanding improves with scale and reasoning, we evaluate the full `Qwen3` family in both non-thinking and thinking modes on the main tasks. Results in Appendix H show consistent gains with model size and a clear boost from thinking mode, while harder tasks (e.g., 2-rhyme RSG) remain challenging even for the largest model.

6 Discussions

In summary, our findings suggest that although LLMs possess extensive phonological knowledge, they generally underperform on tasks requiring phonological understanding, likely due to limited intrinsic phonological associations between words and difficulty in leveraging phonological knowledge. However, reasoning can elevate performance to a human-comparable level. Based on empiri-

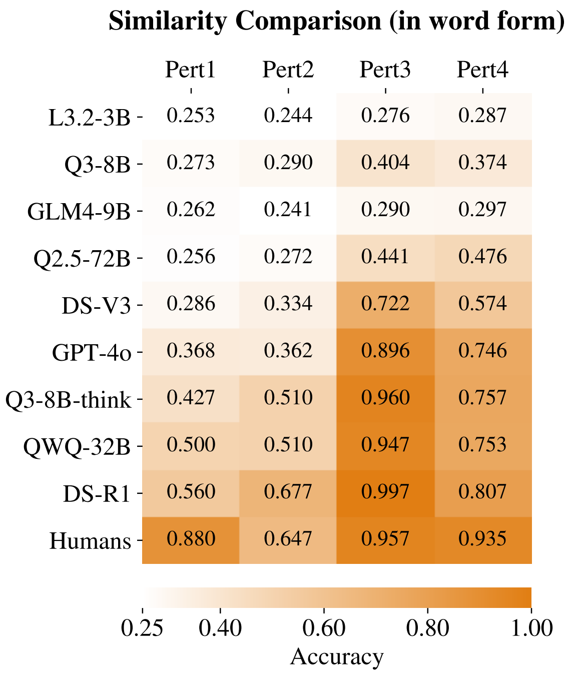

Figure 5: Results of the SC task in word form. L, Q and DS denote `Llama`, `Qwen` and `DeepSeek`, respectively.

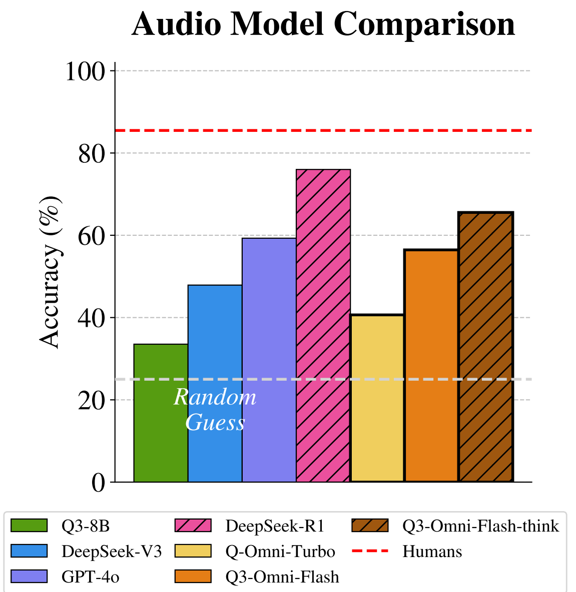

Figure 6: Average Similarity Comparison (SC) results of audio models across four perturbation types (PTs), compared with text-only LLMs.

cal analysis, we further propose a hypothesis for the underlying mechanism of LLM's phonological understanding, i.e., reasoning via the sound layer (§ 5.2.2), which can be interpreted as a temporary construction of phonological associations.

Compared to text-only LLMs, the audio model `Qwen3-Omni-Flash-think` performs better under certain perturbation settings and worse under others (see Figure 11), suggesting a distinct pattern of phonological understanding. However, inspection of reasoning traces indicates that many instances still rely on transcribing audio and reasoning over pinyin, implying a predominantly text-centric pipeline. How multimodality contributes to phonological judgments remains an open question.

Finally, scaling analysis reveals consistent improvements with increasing model size and a clear boost from enabling reasoning, suggesting that phonological understanding benefits from both scale and reasoning. Nevertheless, even the largest model in thinking mode continues to underperform on certain tasks, indicating substantial room for future research.

7 Conclusion

In this work, we introduce the concept of *phonological understanding* and propose criteria for its evaluation. To this end, we present `Phun-Bench`, a purpose-built Chinese benchmark with various tasks and settings, and assess LLMs across three dimensions: Homophony, Rhyme and Phonetic Similarity. Results show that despite their extensive phonological knowledge, LLMs generally fall short in tasks requiring phonological understanding, while performance scales with model size and is further enhanced by reasoning. We further analyze the *challenges* faced by LLMs and propose a hypothesis regarding the underlying *mechanism* of their phonological understanding, offering insights for related fields and future research. Finally, our exploration of audio models underscores a promising multimodal direction for advancing phonological tasks.

Acknowledgements

We appreciate the anonymous reviewers for their valuable comments. This work was supported by the National Natural Science Foundation of China (No. 62376245), National Key Research and Development Project of China (No. 2018AAA0101900), and MOE Engineering Research Center of Digital Library.

Limitations

First, our benchmark and analysis focus only on Chinese. While we argue that Chinese is a strong case for evaluating phonological understanding (as discussed in § 3.2), LLMs may behave differently in other languages. Because each language has its own phonological system and characteristics, directly transferring our benchmark design is generally unsuitable. Extending Phun-Bench to other languages remains an important direction for future work. We acknowledge the potential overgeneralization of the mechanism of phonological understanding we propose. For languages with an alphabetic writing system and more regular grapheme–phoneme mapping (e.g., Italian), explicitly conversion to the sound layer may not be necessary.

Second, we evaluate only publicly available models (see Appendix D.1) without post-training specifically on phonological tasks. Such task-specific fine-tuning could likely improve performance. Nevertheless, we believe that the mechanisms of phonological understanding revealed here would still hold for fine-tuned models.

Finally, our data samples are processed using toolkits and algorithms, followed by manual correction. Although we carefully validated them, minor errors may remain. Similarly, the human baseline we report is based on relatively small samples, and should be viewed as indicative rather than definitive.

Ethical Considerations

We manually reviewed the collected pun sentences and lyrics, removing any potentially offensive content, and do not identify significant risks in our work. Human participants were fairly compensated, receiving 80 CNY for approximately 70 minutes of evaluation. We leverage existing artifacts, i.e., toolkits (jieba, pypinyin) and datasets (chinese-idiom-db, ChineseLyrics, news2016zh, DuanzAI), and believe our usage aligns with their intended research purposes.

References

John R. Anderson, Michael Matessa, and Christian Lebiere. 1997. Act-R: A Theory of Higher Level Cognition and Its Relation to Visual Attention. *Human-Computer Interaction*, 12(4):439–462.

Jonas Belouadi and Steffen Eger. 2023. ByGPT5: End-to-end style-conditioned poetry generation with token-free language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7364–7381, Toronto, Canada. Association for Computational Linguistics.

Benjamin S Bloom, Max D Engelhart, Edward J Furst, Walker H Hill, and David R Krathwohl. 1956. *Taxonomy of educational objectives: The classification of educational goals. Handbook 1: Cognitive domain*. Longman, New York.

Yang Chen, Chong Yang, Tu Hu, Xinhao Chen, Man Lan, Li Cai, Xinlin Zhuang, Xuan Lin, Xin Lu, and Aimin Zhou. 2024a. Are U a joke master? pun generation via multi-stage curriculum learning towards a humor LLM. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 878–890, Bangkok, Thailand. Association for Computational Linguistics.

Yanran Chen, Hannes Gröner, Sina Zarrieß, and Steffen Eger. 2024b. Evaluating diversity in automatic poetry generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19671–19692, Miami, Florida, USA. Association for Computational Linguistics.

Yuyan Chen, Zhixu Li, Jiaqing Liang, Yanghua Xiao, Bang Liu, and Yunwen Chen. 2023. Can Pre-trained Language Models Understand Chinese Humor? In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, pages 465–480. ACM.

Shiyang Cheng, Pengcheng Zhu, Jueting Liu, and Zehua Wang. 2024. A Survey of Grapheme-to-Phoneme Conversion Methods. *Applied Sciences*, 14(24):11790.

Geoffrey Cideron, Sertan Girgin, Mauro Verzetti, Damien Vincent, Matej Kastelic, Zalán Borsos, Brian McWilliams, Victor Ungureanu, Olivier Bachem, Olivier Pietquin, Matthieu Geist, Léonard Hussenot, Neil Zeghidour, and Andrea Agostinelli. 2024. Musicrl: Aligning Music Generation to Human Preferences. In *International Conference on Machine Learning (ICML)*.

CMU. Carnegie mellon university pronouncing dictionary [online].

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Jun-Mei Song, Ruoyu Zhang, R. Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, and 1 others. 2025. Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv.org*, abs/2501.12948.

DeepSeek-AI, A. Liu, Bei Feng, Bing Xue, Bing-Li Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, C. Deng, Chenyu Zhang, and 1 others. 2024. Deepseek-V3 Technical Report. *arXiv.org*, abs/2412.19437.

Avia Efrat, Or Honovich, and Omer Levy. 2023. LMentry: A language model benchmark of elementary language tasks. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 10476–10501, Toronto, Canada. Association for Computational Linguistics.

Anjie Fang, Simone Filice, Nut Limsopatham, and Oleg Rokhlenko. 2020. Using Phoneme Representations to Build Predictive Models Robust to ASR Errors.

In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 699–708. ACM.

J. R. Firch. 1957. *Studies in Linguistic Analysis*, volume 1. Wiley-Blackwell.

Tairan Fu, Raquel Ferrando, Javier Conde, Carlos Arriaga, and Pedro Reviriego. 2024. Why Do Large Language Models (LLMs) Struggle to Count Letters? *arXiv.org*, abs/2412.18626.

Pablo Gervás. 2001. An expert system for the composition of formal Spanish poetry. *Knowledge-Based Systems*, 14(3-4):181–188.

Ariel Goldstein and Gabriel Stanovsky. 2024. Do zombies understand? a choose-your-own-adventure exploration of machine cognition. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 7137–7143, Bangkok, Thailand. Association for Computational Linguistics.

Kyle Gorman, Lucas F.E. Ashby, Aaron Goyzueta, Arya McCarthy, Shijie Wu, and Daniel You. 2020. The SIGMORPHON 2020 shared task on multilingual grapheme-to-phoneme conversion. In *Proceedings of the 17th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pages 40–50, Online. Association for Computational Linguistics.

Ruiqi He, Yushu He, Longju Bai, Jiarui Liu, Zhenjie Sun, Zenghao Tang, He Wang, Hanchen Xia, Rada Mihalcea, and Naihao Deng. 2025. Chumor 2.0: Towards better benchmarking Chinese humor understanding from (ruo zhi ba). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21799–21818, Vienna, Austria. Association for Computational Linguistics.

OpenAI Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o System Card. *arXiv.org*, abs/2410.21276.

Jaehyung Kim, Jaehyun Nam, Sangwoo Mo, Jongjin Park, Sang-Woo Lee, Minjoon Seo, Jung-Woo Ha, and Jinwoo Shin. 2024. Sure: Summarizing Retrievals using Answer Candidates for Open-domain QA of LLMs. In *International Conference on Learning Representations (ICLR)*.

SungHo Kim, Nayeon Kim, Taehee Jeon, and SangKeun Lee. 2025. Polishing every facet of the GEM: Testing linguistic competence of LLMs and humans in Korean. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9955–9984, Vienna, Austria. Association for Computational Linguistics.

Yanis Labrak, Markus Frohmann, Gabriel Meseguer-Brocal, and Elena V. Epure. 2025. Synthetic lyrics detection across languages and genres. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP 2025)*, pages 524–541, Albuquerque, New Mexico. Association for Computational Linguistics.

Davor Lauc. 2025. Polyipa - Multilingual Phoneme-to-Grapheme Conversion Model. In *Natural Language Processing, Information Retrieval and AI Trends 2025*, volume abs/2412.09102, pages 143–156. Academy & Industry Research Collaboration Center.

Jackson L. Lee, Lucas F. E. Ashby, M. Elizabeth Garza, Yeonju Lee-Sikka, Sean Miller, Alan Wong, Arya D. McCarthy, and Kyle Gorman. 2020. Massively Multilingual Pronunciation Modeling with WikiPron. In *International Conference on Language Resources and Evaluation (LREC)*.

Shunwei Lei, Yaoxun Xu, Zhiwei Lin, Huaicheng Zhang, Wei Tan, Hangting Chen, Jianwei Yu, Yixuan Zhang, Chenyu Yang, Haina Zhu, Shuai Wang, Zhiyong Wu, and Dong Yu. 2025. Levo: High-Quality Song Generation with Multi-Preference Alignment. *arXiv.org*, abs/2506.07520.

Jiahao Li, Quan Wang, Zhendong Mao, Junbo Guo, Yanyan Yang, and Yongdong Zhang. 2022. Improving Chinese spelling check by character pronunciation prediction: The effects of adaptivity and granularity. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4275–4286, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Zihong Liang, Xiaojun Quan, and Qifan Wang. 2023. Disentangled phonetic representation for Chinese spelling correction. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13509–13521, Toronto, Canada. Association for Computational Linguistics.

Tyler Loakman, William Thorne, and Chenghua Lin. 2025. Comparing Apples to Oranges: A Dataset & Analysis of LLM Humour Understanding from Traditional Puns to Topical Jokes. *arXiv*.

Junyu Lu, Bo Xu, Xiaokun Zhang, Changrong Min, Liang Yang, and Hongfei Lin. 2023. Facilitating fine-grained detection of Chinese toxic language: Hierarchical taxonomy, resources, and benchmarks. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 16235–16250, Toronto, Canada. Association for Computational Linguistics.

Jianfei Ma, Zhaoxin Feng, Huacheng Song, Emmanuele Chersoni, and Zheng Chen. 2025. Reasoning or memorization? investigating LLMs' capability in restoring Chinese Internet homophones. In *Proceedings of the 3rd Workshop on Towards Knowledgeable Foundation Models (KnowFM)*, pages 120–139, Vienna, Austria. Association for Computational Linguistics.

Meta AI. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models [online]. 2024.

Ministry of Education and State Language Commission. 2013. *List of Commonly Used Standard Chinese Characters*. The Commercial Press.

Luca Moroni, Javier Aula-Blasco, Simone Conia, Irene Baucells, Naiara Perez, Silvia Paniagua Suárez, Anna Sallés, Malte Ostendorff, Júlia Falcão, Guijin Son, Aitor Gonzalez-Agirre, Roberto Navigli, and Marta Villegas. 2025. Multi-LMentry: Can multilingual LLMs solve elementary tasks across languages? In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34126–34157, Suzhou, China. Association for Computational Linguistics.

Guanyi Mou, Pengyi Ye, and Kyumin Lee. 2020. Swe2: Subword Enriched and Significant Word Emphasized Framework for Hate Speech Detection. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 1145–1154. ACM.

Ziqian Ning, Huakang Chen, Yuepeng Jiang, Chunbo Hao, Guobin Ma, Shuai Wang, Jixun Yao, and Lei Xie. 2025. Diffrhythm: Blazingly Fast and Embarrassingly Simple End-to-End Full-Length Song Generation with Latent Diffusion. *arXiv*.

open-dict data. ipa-dict - monolingual wordlists with pronunciation information in ipa [online].

OpenAI. 2025. Openai o3 and o4-mini System Card.

Aitor Ormazabal, Mikel Artetxe, Manex Agirrezabal, Aitor Soroa, and Eneko Agirre. 2022. PoeLM: A meter- and rhyme-controllable language model for unsupervised poetry generation. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3655–3670, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Kyubyong Park and Seanie Lee. 2020. g2pm: A Neural Grapheme-to-Phoneme Conversion Package for Mandarin Chinese Based on a New Open Benchmark Dataset. In *Conference of the International Speech Communication Association (INTERSPEECH)*, pages 1723–1727.

Brian Porter and Edouard Machery. 2024. Ai-generated poetry is indistinguishable from human-written poetry and is rated more favorably. *Scientific Reports*, 14(1).

Qwen Team. Qwq-32b: Embracing the power of reinforcement learning [online]. 2025.

Yesian Rohn. 2024. Duanzai: Slang-Enhanced LLM with Prompt for Humor Understanding. *arXiv.org*, abs/2405.15818.

Hyopil Shin, Sangah Lee, Dongjun Jang, Wooseok Song, Jaeyoon Kim, Chaeyoung Oh, Hyemi Jo, Youngchae Ahn, Sihyun Oh, Hyohyeong Chang, and 1 others. 2025. Kobalt: Korean Benchmark For Advanced Linguistic Tasks. *arXiv.org*, abs/2505.16125.

State Language Commission. 2008. *List of Frequently Used Words in Modern Chinese*, volume 1. The Commercial Press.

Yuchen Su, Yonghua Zhu, Ruofan Wang, Zijian Huang, Diana Benavides-Prado, and Michael Witbrock. 2025. A Survey of Pun Generation: Datasets, Evaluations and Methodologies. *arXiv.org*, abs/2507.04793.

Yusen Sun, Liangyou Li, Qun Liu, and Dit-Yan Yeung. 2023. SongRewriter: A Chinese song rewriting system with controllable content and rhyme scheme. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 12863–12880, Toronto, Canada. Association for Computational Linguistics.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. PhonologyBench: Evaluating phonological skills of large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pages 1–14, Bangkok, Thailand. Association for Computational Linguistics.

Melanie Walsh, Anna Preus, and Maria Antoniak. 2024. Sonnet or not, bot? poetry evaluation for large models and datasets. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15568–15603, Miami, Florida, USA. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Jörg Wöckener, Thomas Haider, Tristan Miller, The-Khang Nguyen, Thanh Tung Linh Nguyen, Minh Vu Pham, Jonas Belouadi, and Steffen Eger. 2021. End-to-end style-conditioned poetry generation: What does it take to learn from examples alone? In *Proceedings of the 5th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature*, pages 57–66, Punta Cana, Dominican Republic (online). Association for Computational Linguistics.

Xiaofeng Wu, Karl Stratos, and Wei Xu. 2025. The impact of visual information in Chinese characters: Evaluating large models' ability to recognize and utilize radicals. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 331–350, Albuquerque, New Mexico. Association for Computational Linguistics.

Yunze Xiao, Yujia Hu, Kenny Tsu Wei Choo, and Roy Ka-Wei Lee. 2024. ToxiCloakCN: Evaluating robustness of offensive language detection in Chinese with cloaking perturbations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6012–6025, Miami, Florida, USA. Association for Computational Linguistics.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, and 1 others. 2025a. Qwen2.5-Omni Technical Report. *arXiv.org*, abs/2503.20215.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xianzhong Shi, Ting He, Xinfa Zhu, and 1 others. 2025b. Qwen3-Omni Technical Report. *arXiv*.

Nan Xu and Xuezhe Ma. 2025. LLM the genius paradox: A linguistic and math expert's struggle with simple word-based counting problems. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3344–3370, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhijun Xu, Siyu Yuan, Lingjie Chen, and Deqing Yang. 2024. "a good pun is its own reword": Can large language models understand puns? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11766–11782, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 Technical Report. *arXiv.org*, abs/2505.09388.

Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, and 1 others. 2024. Qwen2.5 Technical Report. *arXiv.org*, abs/2412.15115.

Chengyue Yu, Lei Zang, Jiaotuan Wang, Chenyi Zhuang, and Jinjie Gu. 2024. CharPoet: A Chinese classical poetry generation system based on token-free LLM. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 315–325, Bangkok, Thailand. Association for Computational Linguistics.

Zhiwei Yu, Hongyu Zang, and Xiaojun Wan. 2020. Homophonic pun generation with lexically constrained rewriting. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2870–2876, Online. Association for Computational Linguistics.

Yixiao Yuan, Yangchen Huang, Yu Ma, Xinjin Li, Zhenglin Li, Yiming Shi, and Huapeng Zhou. 2024. Rhyme-aware Chinese lyric generator based on GPT. *Fourth International Conference on Advanced Algorithms and Neural Networks (AANN 2024)*, 13416:134162P – 134162P–6.

Team Glm Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, and 1 others. 2024. Chatglm: A Family of Large Language Models from GLM-130b to GLM-4 All Tools. *arXiv.org*, abs/2406.12793.

Yichao Zhou, Jyun-Yu Jiang, Jieyu Zhao, Kai-Wei Chang, and Wei Wang. 2020. "the boating store had its best sail ever": Pronunciation-attentive contextualized pun recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 813–822, Online. Association for Computational Linguistics.

Jian Zhu, Cong Zhang, and David Jurgens. 2022. Byt5 model for massively multilingual grapheme-to-phoneme conversion. In *Conference of the International Speech Communication Association (INTERSPEECH)*, pages 446–450.

A Chinese Language Concepts

Pinyin Pinyin (拼音) is the official romanization system adopted in China, where each syllable consists of an optional *initial* (声母, consonant), a *final* (韵母, vowel with an optional nasal ending) and a diacritic indicating the *tone* (声调). For example, the pinyin of the character 高 ("tall") is *gāo*, where *g* is the initial, *ao* is the final, and the macro ¯ denotes the first tone. Alternatively, tones can be simply represented numerically, e.g., *gao1*.

Chengyu Chengyu (成语) are traditional Chinese idiomatic expressions, typically consisting of four characters, that encapsulate rich cultural meanings and moral lessons in a concise form. They remain an essential part of Chinese language and education, valued for their ability to express complex thoughts elegantly and succinctly. For example, the chengyu 刻舟求剑 (*kè zhōu qiú jiàn*, literally "to carve a mark on the boat to seek a lost sword") means *to act foolishly by sticking to fixed methods despite changing circumstances*. For a comprehensive introduction, see https://en.wikipedia.org/wiki/Chengyu.

B Supplementary Related Work

B.1 LLMs' Phonological Ability

We list existing benchmarks that include tasks on LLMs' phonological ability in Table 4. Some benchmarks also contain tasks targeting other aspects (e.g., morphology, semantics), which we omit here. We further omit monolingual G2P benchmarks, as G2P is an established field with numerous dedicated datasets, and including them would distract from comparison.

Benchmark	Year	Phonology-Related Tasks	Language
ipa-Dict (open-dict data)	2016	G2P conversion	multilingual
g2pM (Park and Lee, 2020)	2020	polyphone disambiguation	Chinese
WikiPron (Lee et al., 2020)	2020	G2P conversion	multilingual
SIGMORPHON 2020 Shared Task on Multilingual Grapheme-to-Phoneme (Gorman et al., 2020)	2021	G2P Conversion	multilingual
CharsiuG2P (Zhu et al., 2022)	2022	G2P conversion	multilingual
LMentry (Efrat et al., 2023)	2023	rhyme word judgment, and homophone judgement	English
PhonologyBench (Suvarna et al., 2024)	2024	G2P conversion, syllable counting and rhyme word generation	English
KoBALT (Shin et al., 2025)	2025	Korean phonological rules	Korean
KoGEM (Kim et al., 2025)	2025	Korean phonological rules	Korean
Multi-LMentry (Moroni et al., 2025)	2025	rhyme word judgment and homophone judgement	multilingual
HomoP-CN (Ma et al., 2025)	2025	Internet homophone recovery	Chinese

Table 4: Overview of existing benchmarks evaluating LLMs' phonological ability. Monolingual G2P datasets are omitted, since they form an established field and would distract from comparison.

B.2 Study on Pun

A pun is a form of humorous wordplay that can be broadly categorized into two types: *homographic puns*, which exploit words that share the same spelling but differ in meaning and pronunciation (e.g., *bass* as a fish vs. a musical instrument), and *homophonic puns*, which use word pairs that sound alike but differ in spelling (e.g. *check* and *Czech*). Prior work mainly focuses on three dimensions: pun generation (Su et al., 2025), pun detection (Zhou et al., 2020; Loakman et al., 2025), and pun explanation (Xu et al., 2024). Some studies explicitly leverage pronunciation information for recognizing (Zhou et al., 2020) or generating (Yu et al., 2020) homophonic puns. Chen et al. (2023) construct a Chinese humor dataset and categorize it into three types: harmonic humor (i.e., homophonic puns), ambiguous humor and incongruous humor. Similarly, Rohn (2024) curates a Chinese homophonic pun dataset, `DuanzAI`. These works primarily focus on developing methods for pun generation and only incidentally touch open LLMs' phonological ability. *We adapt* `DuanzAI` *and process it for our Contextual Homophone Recognition (§ 4.2.2) task to evaluate and analyze LLMs' phonological understanding.*

B.3 Rhyme-Related Generation

Rhyme is a key feature in various literary forms, such as poetry and lyrics. Early systems for automatic poetry generation were largely hand-engineered (Gervás, 2001), and neural approaches learn from collections of real poetry data. For example, Wöckener et al. (2021) show that recurrent neural networks (RNNs) struggle to capture sound phenomena such as rhyme and alliteration, and CharPoet (Yu et al., 2024), a Chinese classical poetry generation system, does not handle the rhyme aspect. Later, models such as PoeLM (Ormazabal et al., 2022) and ByGPT5 (Belouadi and Eger, 2023) were developed to generate verses end-to-end, conditioned on stylistic attributes like rhyme and meter. With the advancement of LLMs, poems generated by GPT-3.5 are even rated more favorably than human-written ones in certain aspects such as rhythm (Porter and Machery, 2024). Additionally, Chen et al. (2024b) examine diversity in automatic poetry generation and Walsh et al. (2024) test whether LLMs can identify the poetic form of given poems.

Lyrics generation has received relatively little attention (Sun et al., 2023), with most related work instead focusing on music or multimodal song generation (Lei et al., 2025; Cideron et al., 2024; Ning et al., 2025). Evaluations by Labrak et al. (2025) show that LLM-generated lyrics can be difficult for humans to distinguish from human-written ones.

These studies mainly target poetry and lyrics generation, only tangentially involving rhyme and LLMs' phonological ability. *Building on these insights, we design the Rhyming Sentence Generation task to directly evaluate and analyze LLMs' phonological understanding.*

C Phun-Bench Construction Details and Statistics

C.1 Verification Experiment

Considering that there are more than 80k Chinese characters and most of them are rarely used or variants of others, we curate characters from the *List of Commonly-Used Standard Chinese Characters* (《通用规范汉字表》, Ministry of Education and State Language Commission, 2013). The list is published by the government of China, composed of 8,105 characters, categorized into 3 tiers:

- **Tier 1:** including 3,500 characters designated as frequently used characters.
- **Tier 2:** including 3,000 characters designated as commonly-used characters but less frequently used than those in Tier 1.
- **Tier 3:** including 1,605 characters commonly used as names and terminology.

Since many characters are polyphones and their prescribed pronunciations can vary across different sources, we simply filter them out with the `pypinyin` toolkit. This yields 5,345 unique characters as c_{query} for the char2pinyin task.

We then evaluate both LLMs and humans (for experiment setup, see Appendix D). The results, presented in Table 5, show that LLMs (with the exception of the smallest 3B model) generally outperform humans, exhibiting their extensive phonological knowledge. Since Tier-3 characters are rarely seen even by native speakers, we therefore retain only Tier-1 and Tier-2 characters in Phun-Bench.

C.2 Perturbation Algorithms

We present the pseudocode for the perturbation algorithms used in our data construction.

Model	Commonness		
	Tier-1	Tier-2	Tier-3
`Llama3.2-3B`	84.02	27.87	4.48
`Qwen3-8B`	99.29	91.18	70.83
`GLM4-9B`	99.01	79.09	25.92
`Qwen2.5-72B`	99.76	92.37	59.84
`DeepSeek-V3`	100.00	99.80	82.20
`GPT-4o`	100.00	99.00	74.80
`Qwen3-8B-think`	99.00	91.33	66.00
`QwQ-32B`	98.00	89.33	51.33
`DeepSeek-R1`	100.00	99.67	83.00
Humans	98.00	72.00	46.00

Table 5: Accuracy (%) of the Verification experiment. We verify that LLMs exhibit extensive phonological knowledge of the pronunciations of characters.

Algorithm 1 is employed in the Homophone Recall task (§ 4.2.1), where the character set $\mathcal{C}$ consists of the 5,345 retained monophonic characters (see § 3.3), the chengyu set $\mathcal{W}$ consists of the 2,857 retained four-character chengyu (see § 4.2.1), and $\mathcal{M}_{\text{c2p}}$ is implemented using `pypinyin` for individual characters or the pronunciations provided in the original chengyu dataset (and with manual error corrections for any errors). In lines 3–6, we construct the pinyin-to-character map G. In lines 7–22, we perturb each four-character chengyu in $\mathcal{W}$ if a homophonic substitution exists for every character. The perturbed result w_{ref} is provided to LLMs, with the original chengyu w_{target} serving as the golden answer.

Algorithm 2 is employed in the Similarity Comparison task (§ 4.4), where word set $\mathcal{W}$ consists of the retained 13,741 two-character words, and PT denotes one of the perturbation types defined in Table 1. In lines 7–10, for each reference word w_{ref}, we search for another word w whose two characters match the similarity type specified by PT, aligned with the characters in w_{ref}. We then obtain one answer word w_{close} and three distractor words w_{far}, which are shuffled to construct the four candidate words w_{cand_i}, $i = 1, 2, 3, 4$.

C.3 Statistics

We present the statistics of Phun-Bench in Table 6.

Algorithm 1 Chengyu Perturbation in the Homophone Recall Task

Require: Character set $\mathcal{C}$, chengyu set $\mathcal{W}$, character-to-pinyin converter $\mathcal{M}_{\text{c2p}}$
Ensure: Each character c in $\mathcal{C}$ is monophonic
Initialize empty pinyin-to-character map $G \leftarrow \{\}$
Initialize output list of tuples $\mathcal{T} \leftarrow \varnothing$
▷ Populate G with each character $c \in \mathcal{C}$
for each character c in $\mathcal{C}$ **do**
    $p \leftarrow \mathcal{M}_{\text{c2p}}(c)$
    $G[p] \leftarrow G[p] \cup \{c\}$ ▷ $G[p]$ defaults to $\varnothing$ if p is new
end for
▷ Perturb each chengyu from $\mathcal{W}$
for each target chengyu w_{target} in $\mathcal{W}$ **do**
    $w_{\text{ref\_list}} \leftarrow [\_, \_, \_, \_]$ ▷ Placeholders for 4 characters
    for $i = 0$ to $|w_{\text{target}}| - 1$ **do** ▷ $i = 0, 1, 2, 3$
        $c \leftarrow w[i]$
        $p \leftarrow \mathcal{M}_{\text{c2p}}(c)$
        $c_{\text{homo}} \leftarrow \texttt{sample}(G[p] \setminus \{c\})$
        if $c_{\text{homo}} = \texttt{None}$ **then** ▷ Skip if no available homophonic character
            continue
        else
            $w_{\text{ref\_list}}[i] \leftarrow c_{\text{homo}}$
            $w_{\text{ref}} \leftarrow \texttt{concat}(w_{\text{ref\_list}})$ ▷ Convert list to string
        end if
    end for
    $\mathcal{T} \leftarrow \mathcal{T} \cup (w_{\text{ref}},\ w_{\text{target}})$
end for
return $\mathcal{T}$

Algorithm 2 Word Perturbation in the Similarity Comparison Task

Require: Two-character word set $\mathcal{W}$, perturbation type PT
Ensure: $PT \in \{pert_1,\ pert_2,\ pert_3,\ pert_4\}$
Initialize output list $\mathcal{T} \leftarrow \varnothing$
Extract word-level similarity types for answer and distractors: $WT_{\text{close}},\ WT_{\text{far}} \leftarrow PT_{\text{pert}}$
Initialize $w_{\text{close}} \leftarrow \texttt{None}$
Initialize $w_{\text{far}} \leftarrow \varnothing$
for each reference word w_{ref} in $\mathcal{W}$ **do**
    for each word w in $\mathcal{W} \setminus \{w_{\text{ref}}\}$ **do**
        ▷ Character-level similarity judgment omitted; see Table 1
        if $w_{\text{close}} = \texttt{None}$ **and** $\texttt{get\_type}(w_{\text{ref}},\ w) = WT_{\text{close}}$ **then**
            $w_{\text{close}} \leftarrow w$ ▷ The answer
        else if $|w_{\text{far}}| < 3$ **and** $\texttt{get\_type}(w_{\text{ref}},\ w) = WT_{\text{far}}$ **then**
            $w_{\text{far}} \leftarrow w_{\text{far}} \cup \{w\}$ ▷ The distractors
        end if
        if $w_{\text{close}} \neq \texttt{None}$ **and** $|w_{\text{far}}| \geq 3$ **then**
            $w_{\text{cand}},\ i_{\text{ans}} \leftarrow \texttt{shuffle}(\{w_{\text{close}}\} \cup w_{\text{far}})$ ▷ Shuffle candidates and record the answer index
            $\mathcal{T} \leftarrow \mathcal{T} \cup \{(w_{\text{ref}},\ w_{\text{cand}},\ i_{\text{ans}})\}$
            break
        end if
    end for
end for
return $\mathcal{T}$

Task	Setting	Size
char2pinyin	Tier-1	2,122
	Tier-2	2,085
	Tier-3	1,138
char2char	-	244
pinyin2char		
Homophone Recall	-	1,098
Contextual Homophone Recognition	-	1,684
Rhyming Sentence Generation	1-rhyme	1,000
	2-rhyme	
Similarity Comparison	$pert_1$	945
	$pert_2$	880
	$pert_3$	431
	$pert_4$	2,013

Table 6: Dataset size of Phun-Bench and related tasks.

D Experiment Setup

D.1 Model Implementations

This section lists the models used in our experiments along with their hyperparameters for reproducibility. When available, temperature and top_p values follow the official recommendations for better performance.

- **Llama-3.2-3B-Instruct** (Meta AI, 2024): open-source, non-thinking. temperature = 0.7, top_p = 0.8.
- **Qwen3-8B** (Yang et al., 2025): open-source, evaluated in both non-thinking and thinking modes. Non-thinking mode: temperature = 0.7, top_p = 0.8. Thinking-mode: temperature = 0.6, top_p = 0.95.
- **GLM-4-9B** (Zeng et al., 2024): open-source, non-thinking. temperature = 0.7, top_p = 0.8.
- **Qwen2.5-72B-Instruct-AWQ** (Yang et al., 2024): open-source, non-thinking. temperature = 0.7, top_p = 0.8.
- **QwQ-32B-AWQ** (Qwen Team, 2025): open-source, thinking. temperature = 0.6, top_p = 0.95.
- **DeepSeek-V3-0324** (DeepSeek-AI et al., 2024): open-source, non-thinking, accessed via API. temperature = 1.0, top_p = 0.8.
- **DeepSeek-R1-0528** (DeepSeek-AI et al., 2025): open-source, thinking, accessed via API. temperature = 1.0, top_p = 0.8.
- **GPT-4o-2024-11-20** (Hurst et al., 2024): closed-source, non-thinking, accessed via API. temperature = 1.0, top_p = 0.8.
- **Qwen-Omni-Turbo-2025-03-26**: multimodal, closed-source, non-thinking, accessed via API. Reportedly based on Qwen2.5-Omni[13] (Xu et al., 2025a). Default hyperparameters are used.
- **Qwen3-Omni-Flash-2025-09-15** (Xu et al., 2025b): multimodal, closed-source, evaluated in both non-thinking and thinking modes. We access it through API. Default hyperparameters are used.
- **Qwen-TTS-2025-04-10**[14]: closed-source, accessed via API. Used to provide audio-modal input for multimodal models.

For the scaling analysis on the open-source Qwen3 family models in § 5.5, we access the models via API due to time constraints during the rebuttal phase.

Unless otherwise specified, models are deployed locally via Hugging Face. We use one A6000 GPU for Llama-3.2-3B-Instruct, Qwen3-8B, and GLM-4-9B, and two or four A6000 GPUs for Qwen2.5-72B-Instruct-AWQ and QwQ-32B-AWQ. Inference for non-thinking models takes only minutes per setting. For thinking models, Rhyming Sentence Generation and Similarity Comparison take hours due to thousands of reasoning tokens, while other tasks complete within an hour. As noted in § 5.1, budget constraints limit runs to 300 samples per setting for thinking models, 500 for DeepSeek-V3 and GPT-4o, and the full set for others. For reproducibility, we fix the random seed at 42.

D.2 Human Evaluation

All three human participants invited were native Chinese speakers, undergraduate students majoring

[13] https://www.alibabacloud.com/help/en/model-studio/qwen-omni

[14] https://www.alibabacloud.com/help/en/model-studio/qwen-tts

in clinical medicine, pharmacy, and cultural heritage and museology, respectively, each remunerated with 80 CNY for evaluation. They were not involved in the study design and were informed about data usage before participation. For the char2pinyin verification experiment, we sampled 100 instances from each tier and shuffled them. For the Homophone Recall (HR) and Contextual Homophone Recognition (CHR) tasks, we sampled 100 instances each; for the Similarity Comparison (SC) task, we sampled approximately 50 instances for each perturbation type and shuffled them, across both word-form and pinyin-form settings. The sampled instances were evenly distributed among participants for evaluation, which we estimate took roughly 70 minutes per participant to complete. An instruction is provided for each human-evaluated task (see Appendix I).

Two participants reported the low readability of the SC task in pinyin form, and the performance was significantly lower, ranging from 2% to 12%. Therefore, we only use the word-form results as human baseline.

D.3 Package Version

We use the `jieba` (v0.42.1) and `pypinyin` (v0.55.0) packages for Chinese word segmentation and character-to-pinyin conversion.

E RQ 1: How do LLMs recover the homophonic target word?

E.1 Case study

Figure 12 and 13 present two reasoning traces of HR, generated by `DeepSeek-R1` and `Qwen3-8B-think`. Notably, the LLMs first convert the homophone into its pronunciation (i.e., pinyin) and then attempt to recover the target word, supporting our two-step hypothesis that LLMs perform phonological understanding via the sound layer.

E.2 Character-Level Homophone Recall

We analyze the mechanism of phonological understanding using the Character-Level Homophone Recall (CLHR) task. Following the char2pinyin experiment (§ 5), we use Tier-1 and Tier-2 monophonic characters from the *List of commonly-used Standard Chinese Characters* (Ministry of Education and State Language Commission, 2013) and obtain their pinyin using `pypinyin`. In the char2char setting, LLMs are provided with a character c_{ref} and prompted to recall five homophonic characters c_{target}; in the pinyin2char setting, the pinyin p is provided instead.

We ensure that for each sample, at least five Tier-1 or Tier-2 monophonic characters are available as answers, resulting in 244 samples. Using `pypinyin`, we check if the five output characters $\hat{c}_{\text{target}}$ have the required pronunciation p, assigning a score of 0.2 per correct character (e.g., three correct characters yield 0.6). The average score is then computed across all samples.

The results are presented in Table 7. Following the analysis in § 5.2.2, we examine the performance relationship between the three tasks and find that the proposed inequality (and hence the hypothesis) does not hold consistently. We attribute this to the fact that LLMs can solve these tasks by memorizing dictionary data, leveraging human-curated associations between homophonic characters, so that an indirect phonological understanding via the sound layer is not strictly necessary.

F RQ 2: What Challenges Do LLMs Face in RSG?

We collect 41 failed reasoning traces of the 2-rhyme setting from `DeepSeek-R1` and another 50 from `QwQ-32B`, and manually categorize them into four error types. Their proportions are shown in Figure 7.

- **Indexing Error (IE):** the model fails to correctly identify the last or next-to-last character in either the reference or the generated sentence. For example, rhyming “一直都这样孤独地来去” with “独自面对这苍白的衣白的衣”.
- **Misgrouping:** the model assigns a character to a wrong rhyme group (RG). For example, placing 飘 (*piāo*) into G_{11} (finals *ou*, *iou*), instead of G_{10} (finals *ao*, *iao*).
- **Instruction-Following Error (IFE):** the model fails to comply with the given instruction. For example, generating a 1-rhyme sentence when a 2-rhyme is required.
- **Others:** miscellaneous errors, such as acknowledging potential rhyme violations but still producing the sentence, or generating a final answer different from the decision in the reasoning trace.

For `Qwen3-8B-think`, we observe an abnormal tendency to produce meaningless repetitions of

Task	Setting	Model									Relationship
		L3.2-3B	Q3-8B	GLM4-9B	Q2.5-72B	DS-V3	GPT-4o	Q3-8B-think	QwQ-32B	DS-R1	
CLHR	$P(p \mid c_q)$	0.161	0.809	0.523	0.760	0.999	0.995	0.952	0.937	0.998	$r = \frac{P_1 P_2}{P_3} \geq 1?$
	$P(c_t \mid p)$	0.432	0.776	0.673	0.880	0.991	0.961	0.903	0.932	0.955	
	$P(c_t \mid c_q)$	0.125	0.420	0.388	0.668	0.949	0.917	0.778	0.840	0.959	
	r	0.55	1.50	0.91	1.00	1.04	1.04	1.11	1.04	0.99	

Table 7: The accuracy on char2pinyin, pinyin2char and char2char conversion. L, Q and DS denote `Llama`, `Qwen` and `DeepSeek`, respectively. Note that the inequality doesn't hold consistently.

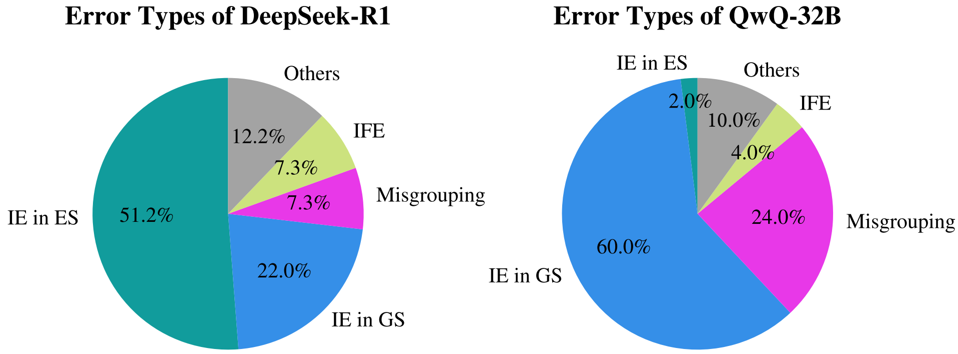

Figure 7: The proportions of error types in the 2-rhyme setting. ES denotes the example sentence; GS denotes the generated sentence.

example sentences, generated sentences, or individual characters/words, across both 1-rhyme and 2-rhyme settings. The model rarely succeeds in genuinely "composing" a rhyming sentence; the few correct cases appear more coincidental than deliberate. The underlying cause for this behavior remains unclear.

G RQ 3: How Do LLMs "Perceive" Phonetic Similarity?

G.1 Case study

Figure 14 and 15 present two reasoning traces of HR, generated by `DeepSeek-R1` and `Qwen3-8B-think`. Notably, the LLMs first convert the homophone into its pronunciation (pinyin) and then attempt to recover the target word, supporting our two-step hypothesis that LLMs perform phonological understanding via the sound layer (as discussed in § 5.2.2).

G.2 Similarity Comparison in Pinyin Form

We present the results of the SC task in pinyin form in Figure 8, and a performance comparison between the two settings in Figure 10. We observe that the pinyin-form setting generally improves LLM performance[15], consistent with our hypothesis that LLMs perform phonological understanding primarily via the sound layer.

G.3 Evaluation on Audio-Enabled LLMs

We use `Qwen-TTS`[16] to convert the questions into audio files. These audio files, combined with a short prompt (see Figure 30), are then evaluated by `Qwen-Omni-Turbo` and `Qwen3-Omni-Flash`. The speech templates and prompts are shown in Figure 29 and 30.

The results for each perturbation setting are shown in Figure 11. Audio capability clearly improves performance; when combined with explicit reasoning, `Qwen3-Omni-Flash-think` even surpasses DeepSeek-R1 in the $Pert_4$ setting. Interestingly, the relative strength of audio models differs from that of text-only LLMs: for example, LLMs perform better in the $Pert_3$ setting, likely because token-level similarity makes tone differences easy to capture in text form. In contrast, audio models excel in the $Pert_4$ setting, which is relatively easier to resolve acoustically, as also testified by human performance.

[15] In contrast, human participants' performance drops, as they reported that the pinyin-form setting was difficult to read.

[16] https://www.alibabacloud.com/help/en/model-studio/qwen-tts

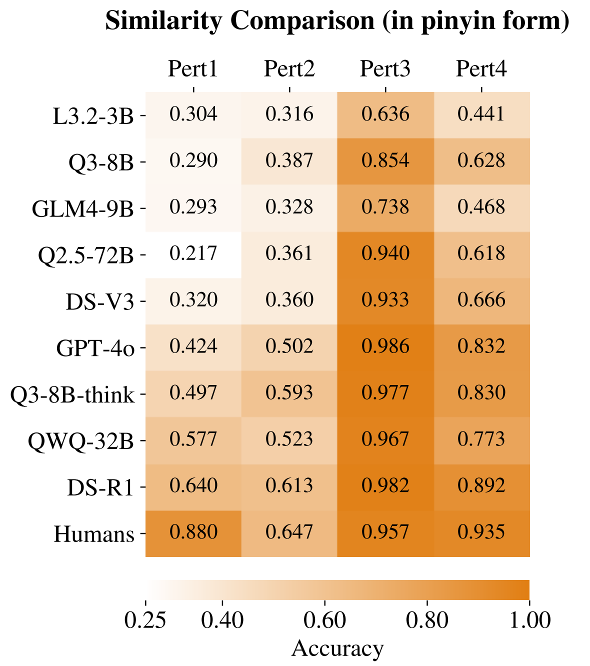

Figure 8: Results of the SC task in pinyin form. L, Q and DS denote `Llama`, `Qwen` and `DeepSeek`, respectively. For the human baseline, we still use results from the word-form setting.

H Scaling and Mode Analysis of Qwen3 Models

To systematically analyze how phonological understanding emerges and scales with model size and reasoning mode, we conduct evaluations of our main tasks on the open-sourced `Qwen3` models, covering all eight available sizes/architectures (from 0.6B to 235B-A22B), in both non-thinking and thinking modes.

The results, summarized in Table 8 and Figure 9, demonstrate that:

- **Performance scales with model size:** Across nearly all tasks, larger models consistently outperform smaller ones. For example, in the Similarity Comparison (SC) task, smaller models (0.6B and 1.7B) achieve near-random performance (~ 0.25) on $Pert_4$, but the capability clearly emerges as model size increases.

- **Benefit of long CoT (thinking mode):** The thinking mode significantly outperforms the non-thinking mode, demonstrating the clear benefit of explicit step-by-step reasoning via the sound layer.

Despite these improvements, even the largest model (235B-A22B) in thinking mode still shows room for improvement on certain challenging tasks, such as 2-rhyme Rhyming Sentence Generation (RSG).

I Prompt Templates and Human Instructions

The prompt templates used in model evaluations are as follows:

- **Verification Experiment**: See Figure 16.
- **Homophony-Based Understanding**: See Figure 17 and 18.
- **RQ 1**: See Figure 19 to 24.
- **Rhyme-Based Understanding and RQ 2:** See Figure 25 and 26.
- **Phonetic-Similarity-Based Understanding and RQ 3:** See Figure 27 and 28.

The speech and prompt templates used in the audio-model experiment (§ 5.4.3) are in Figure 29 and 30.

The human instructions are shown in Figures 31 to 35, and a screenshot of an `.xlsx` file provided to participants is shown in Figure 36.

Task	Non-Thinking Mode								Thinking Mode							
	0.6B	**1.7B**	**4B**	**8B**	**14B**	**32B**	**30B-A3B**	**235B-A22B**	**0.6B**	**1.7B**	**4B**	**8B**	**14B**	**32B**	**30B-A3B**	**235B-A22B**
HR	0.00	0.00	0.08	0.14	0.23	0.24	0.25	0.37	0.00	0.08	0.41	0.67	0.78	0.76	0.77	0.98
CHR (DR)	0.80	0.90	0.88	0.96	0.96	0.96	0.97	0.98	0.71	0.91	0.96	0.96	0.99	0.97	0.97	0.99
CHR (RR)	0.05	0.12	0.23	0.35	0.42	0.45	0.41	0.54	0.07	0.15	0.22	0.36	0.53	0.53	0.47	0.70
RSG (1-Rhyme)	0.07	0.10	0.16	0.39	0.49	0.60	0.40	0.56	0.13	0.14	0.48	0.65	0.77	0.74	0.71	0.78
RSG (2-Rhyme)	0.01	0.01	0.01	0.03	0.04	0.05	0.04	0.04	0.02	0.02	0.10	0.07	0.31	0.18	0.10	0.24
SC ($Pert_1$)	0.23	0.25	0.25	0.27	0.26	0.28	0.26	0.27	0.27	0.31	0.38	0.38	0.41	0.48	0.45	0.45
SC ($Pert_2$)	0.23	0.25	0.27	0.29	0.30	0.34	0.29	0.34	0.25	0.26	0.38	0.60	0.50	0.56	0.53	0.49
SC ($Pert_3$)	0.29	0.30	0.39	0.41	0.49	0.54	0.40	0.63	0.33	0.55	0.71	0.97	0.91	0.96	0.91	0.99
SC ($Pert_4$)	0.27	0.28	0.30	0.38	0.47	0.54	0.42	0.58	0.25	0.42	0.51	0.79	0.76	0.74	0.71	0.79
SC (Ave.)	0.25	0.27	0.30	0.34	0.38	0.42	0.34	0.45	0.27	0.38	0.49	0.68	0.65	0.68	0.65	0.68
Overall	0.09	0.12	0.19	0.31	0.38	0.43	0.35	0.48	0.12	0.19	0.40	0.59	0.68	0.68	0.65	0.78

Table 8: Performance scaling of `Qwen3` models across different sizes and modes. The overall score is computed as the average of HR, CHR (RR), RSG (1-Rhyme), RSG (2-Rhyme), and SC (Ave.).

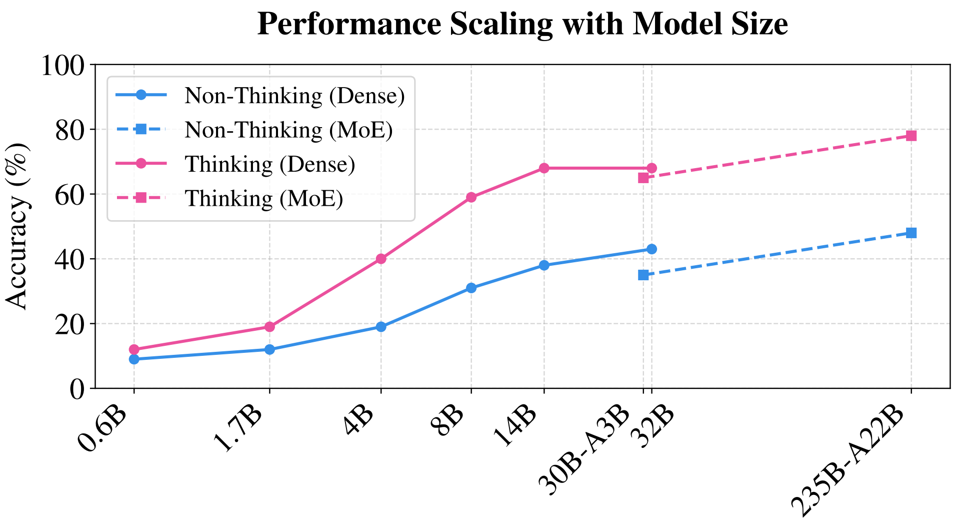

Figure 9: Overall performance scaling of `Qwen3` models across different sizes (0.6B to 235B-A22B) and modes (Non-Thinking vs. Thinking).

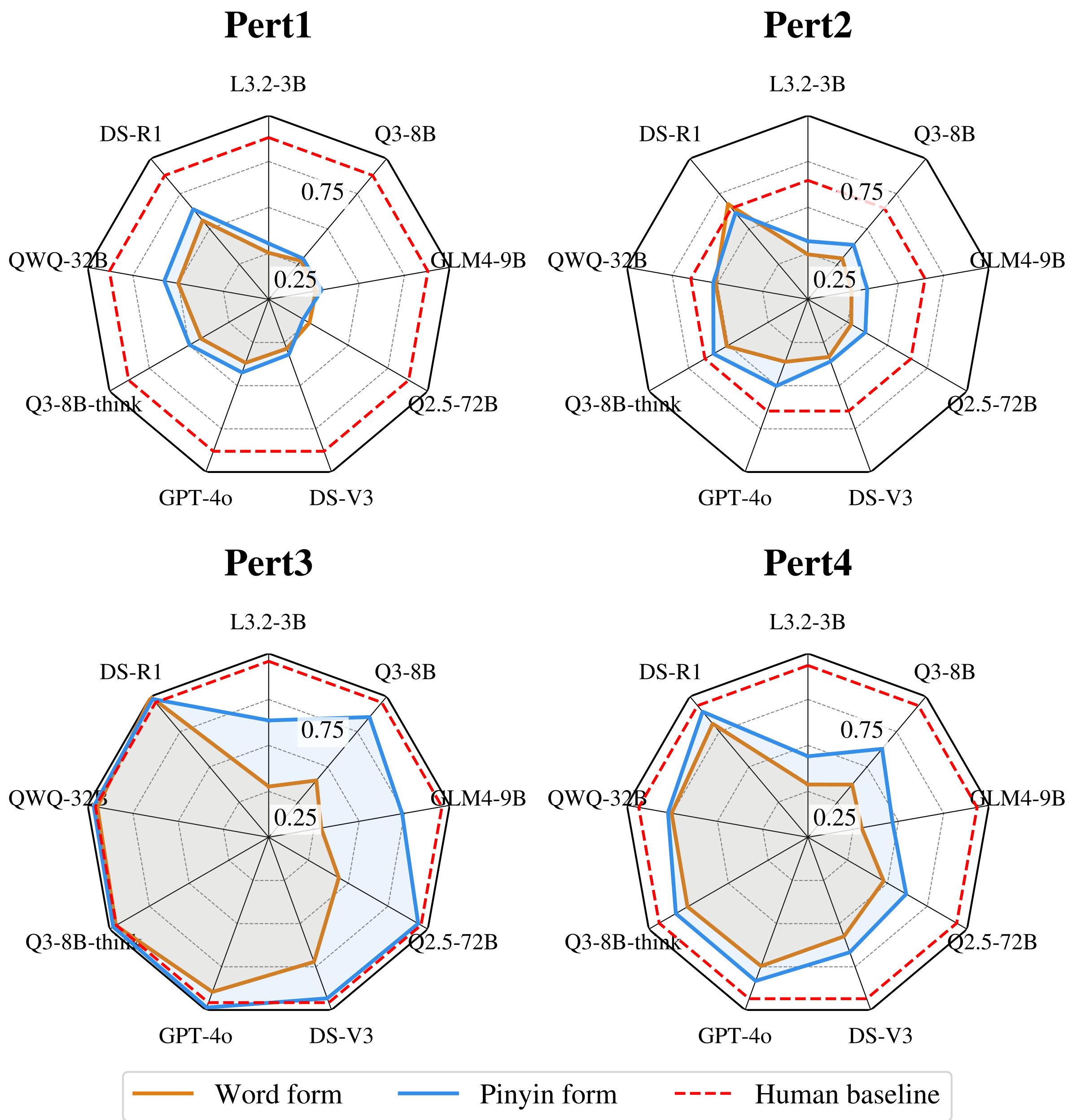

Figure 10: Comparison between performances under the pinyin-form and word-form settings. L, Q and DS denote `Llama`, `Qwen` and `DeepSeek`, respectively.

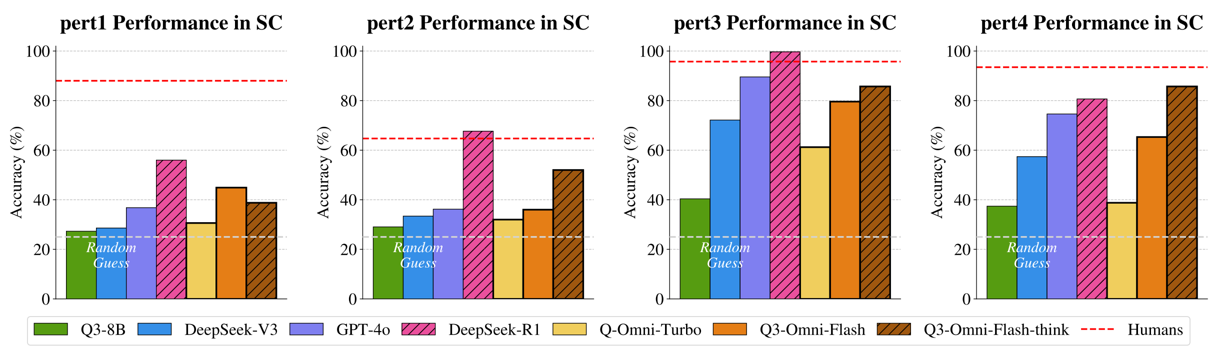

Figure 11: Similarity Comparison results for audio models across perturbation settings.

Case Study on Homophone Recall (Correct)

Prompt:

你是一位精通汉语的大师。请根据下面给出的一个“短语”，猜出和它在读音上最相近的四字成语（不要考虑语义），以json格式输出答案。示例如下：

```
<content>孰语盆笔</content>
输出：{
    "answer": "无与伦比"
}
<content>律师捕霜</content>
输出：{
    "answer": "屡试不爽"
}
<content>王杨捕劳</content>
输出：{
    "answer": "亡羊补牢"
}
本题目给出的短语如下：
<content>呻鳞骑竟</content>
```

Thought:

首先，用户提供了短语："呻鳞骑竟"。我需要根据它的读音猜出最相近的四字成语，只考虑读音，不考虑语义。输出必须是JSON格式，如示例所示。

(... analysis on examples omitted)

现在，我的短语是 "呻鳞骑竟"。我需要找出它在读音上最接近的四字成语。
先分解 "呻鳞骑竟" 的发音：
- "呻"：拼音是 "shēn"（shēn）。
- "鳞"：拼音是 "lín"（lín）。
- "骑"：拼音是 "qí"（qí）。
- "竟"：拼音是 "jìng"（jìng）。

所以，整个短语的发音是 "shēn lín qí jìng"。

我需要找到一个四字成语，其发音最接近 "shēn lín qí jìng"。
回想常见的四字成语：

- 可能成语有 "身临其境"? "身临其境" 发音是 "shēn lín qí jìng"。

(... verification and more guesses omitted)

现在，输出格式必须是JSON：
```
{
    "answer": "身临其境"
}
```
在示例中，是完整的JSON对象。
最后，确保只输出JSON，没有其他文本。

Answer: "身临其境"

Figure 12: A correct reasoning trace for the Homophone Recall task from DeepSeek-R1.

Case Study on Homophone Recall (Incorrect)

Prompt:

你是一位精通汉语的大师。请根据下面给出的一个“短语”，猜出和它在读音上最相近的四字成语（不要考虑语义），以json格式输出答案。示例如下：

<content>孰语盆笔</content>
输出：{
    "answer": "无与伦比"
}
<content>律师捕霜</content>
输出：{
    "answer": "屡试不爽"
}
<content>王杨捕劳</content>
输出：{
    "answer": "亡羊补牢"
}
本题目给出的短语如下：
<content>案怖臼颁</content>

Thought:

好的，我现在需要解决用户的问题，根据给出的拼音“案怖臼颁”来猜测一个读音相近的四字成语。首先，我需要将每个拼音拆解，然后寻找对应的汉字，再组合成成语。
首先，分析每个拼音对应的可能汉字：
1．“案”拼音是an，可能的汉字有案、安、暗等。
2．“怖”拼音是bu，可能的汉字有怖、布、步等。
3．“臼”拼音是jiu，可能的汉字有臼、旧、就等。
4．“颁”拼音是ban，可能的汉字有颁、班、搬等。
接下来，我需要将这些可能的汉字组合成四字成语，并且每个字的拼音要与原短语中的拼音一致。例如，第一个字是an，第二个是bu，第三个是jiu，第四个是ban。那么可能的组合是：安布旧班？但这样组合起来似乎没有意义，可能需要调整每个字的可能选项。
或者，可能每个拼音对应的字不是直接对应，而是每个字的拼音与原短语中的拼音相同，但字形不同。例如，“案”可能对应“安”，“怖”对应“布”，“臼”对应“旧”，“颁”对应“班”，这样组合成“安布旧班”，但这样的成语不存在。
可能我需要重新考虑每个拼音对应的字。例如，“案”可能对应“按”，但拼音是an，所以可能没问题。或者“怖”可能对应“捕”（bu），但拼音是bu，所以可能。然后“臼”可能对应“就”（jiu），而“颁”对应“班”（ban）。这样组合起来是“按捕就班”？这似乎也不对。

或者，可能每个字的拼音完全对应，但需要找到四字成语。例如，原短语是an、bu、jiu、ban。有没有成语的拼音是an bu jiu ban？比如“安步就班”？安步就班的拼音是an bu jiu ban，正好对应。所以答案可能是“安步就班”。

检查一下这个成语是否存在。“安步就班”确实是一个成语，意思是安步当车，就班取舍，形容安详从容，不慌不忙。所以这应该符合要求。因此，正确的答案应该是“安步就班”。

Answer: "安步就班"

Figure 13: An incorrect reasoning trace for the Homophone Recall task from `Qwen3-8B-think`. The correct answer is “按部就班”.

Case Study on Similarity Comparison (Correct)

Prompt:
你是一位精通汉语的大师。请从四个选项中，选出在读音上和给出的词语最接近的一项（不要考虑语义），以json格式输出答案。示例如下：
(... examples omitted)
本题目给出的词语和选项如下：
<content>摊位</content>
<choices>A. 环卫 B. 调味 C. 因为 D. 单位</choices>

Thought:
首先，用户要求我选出在读音上和给出的词语“摊位”最接近的一项。不要考虑语义，只关注发音。输出必须是JSON格式，如{"answer": "选项"}。
给出的词语是“摊位”，拼音是“tān wèi”。
现在，分析每个选项的拼音：
- A. 环卫：拼音是“huán wèi”。
- B. 调味：拼音是“tiáo wèi”。
- C. 因为：拼音是“yīn wèi”。
- D. 单位：拼音是“dān wèi”。
目标词语“摊位”的拼音是“tān wèi”。我需要找出哪个选项的发音最接近“tān wèi”。
比较每个选项：
- A. 环卫：“huán wèi” – 声母“h” vs “t”，韵母“uán” vs “ān”，完全不同。韵尾“wèi”相同，但开头差异大。
- B. 调味：“tiáo wèi” – 声母“t”相同，但韵母“iáo” vs “ān”。韵尾“wèi”相同，但“iáo”和“ān”发音不同。
- C. 因为：“yīn wèi” – 声母“y” vs “t”，韵母“īn” vs “ān”。韵尾“wèi”相同，但开头差异大。
- D. 单位：“dān wèi” – 声母“d” vs “t”，韵母“ān”相同，韵尾“wèi”相同。“d”和“t”都是舌尖音，但“d”是浊音，“t”是清音。在汉语中，“d”和“t”发音相似，都是齿龈塞音，一个浊一个清，但音位接近。韵母“ān”完全一样。
重点比较“摊位”(tān wèi) 和 “单位”(dān wèi)：
- 声母：t vs d – 都是齿龈塞音，t是送气清音，d是不送气浊音。在普通话中，它们发音相似，容易混淆，尤其在某些口音中。
- 韵母：ān 完全相同。
- 韵尾：wèi 完全相同。
所以，“摊位”和“单位”在发音上非常接近，主要区别在声母t和d。
现在看其他选项：
- B. 调味 (tiáo wèi)：声母t相同，但韵母iáo vs ān，差异更大。
- A. 环卫 (huán wèi)：声母和韵母都不同。
- C. 因为 (yīn wèi)：声母和韵母不同。
因此，D. 单位 是最接近的。
回顾之前的例子：
- 在“增添”的例子中，选了“春天”，因为“增添”(zēng tiān) 和 “春天”(chūn tiān) 都有“tiān”，尽管声母不同。
- 在“渐渐”的例子中，选了“见面”，因为“渐渐”(jiàn jiàn) 和 “见面”(jiàn miàn) 共享“jiàn”。
类似地，这里“摊位”和“单位”共享“wèi”，并且“ān”相同，声母t和d相似。
在汉语中，“摊位”读作 tān wèi，“单位”读作 dān wèi。发音上，t和d在音位上对立，但音值接近，尤其在不强调时。
我认为D是最佳选择。
确认输出格式：必须是JSON，如{"answer": "D"}。
所以，我的输出应该是{"answer": "D"}。

Answer: "D"

Figure 14: A correct reasoning trace for the Similarity Comparison task ($Pert_1$ setting) from DeepSeek-R1.

Case Study on Similarity Comparison (Incorrect)

Prompt:
你是一位精通汉语的大师。请从四个选项中，选出在读音上和给出的词语最接近的一项（不要考虑语义），以json格式输出答案。示例如下：
(... examples omitted)
本题目给出的词语和选项如下：
<content>今生</content>
<choices>A. 晋升 B. 发生 C. 民生 D. 寄生</choices>

Thought:
好的，我现在需要解决这个题目，找出与“今生”在读音上最接近的选项。首先，我要仔细分析每个选项的拼音，然后与“今生”的拼音进行比较。
首先，“今生”的拼音是“jīn shēng”。接下来，我要逐个分析选项：

选项A是“晋升”，拼音是“jìn shēn”。这里要注意的是，“晋升”的“晋”发音是jìn，而“今生”的“今”是jīn，这两个韵母不同，声调也不同（“今”是第一声，“晋”是第四声）。所以A选项的发音和“今生”不太接近。

选项B是“发生”，拼音是“fā shēng”。“发生”的“发”是fā，而“今生”的“今”是jīn，声母不同，韵母部分“生”是shēng，而“今生”的“生”也是shēng，但前面的声母不同，所以整体发音差异较大。
选项C是“民生”，拼音是“mín shēng”。“民生”的“民”是mín，而“今生”的“今”是jīn，声母不同，但韵母都是“in”结尾，不过“今生”的“生”是shēng，而“民生”的“生”是shēng，所以韵母部分相同，但声母不同。不过可能需要考虑整体发音的接近程度。
选项D是“寄生”，拼音是“jì shēng”。“寄生”的“寄”是jì，而“今生”的“今”是jīn，声母都是j，但韵母不同，“今”是in，“寄”是i，不过声调方面，“今”是第一声，“寄”是第四声。不过“生”部分都是shēng，所以后半部分相同。
现在比较各个选项与“今生”的发音接近程度。首先看声母部分，“今生”的声母是j（jīn）和sh（shēng）。选项C的“民生”声母是m和sh，选项D的“寄生”是j和sh。所以D选项的声母部分与“今生”的前一个字“今”（j）相同，而“生”部分相同，所以整体发音更接近。而选项C的“民”是m，与j不同。因此，D选项的“寄生”可能更接近。
不过需要再仔细检查。例如，“今生”是jīn shēng，而“寄生”是jì shēng。前一个字的韵母是in和i，声调不同，但声母相同。后一个字完全相同。因此，可能D选项的发音更接近，因为前一个字的声母相同，后一个字完全相同，尽管韵母和声调不同，但整体可能更接近。而选项C的“民生”是mín shēng，前一个字声母不同，但韵母相同，但声母差异较大。因此，D选项更接近。

Answer: "D"

Figure 15: An incorrect reasoning trace for the Similarity Comparison task ($Pert_3$ setting) from `Qwen3-8B-think`. The correct answer is “A”.

Prompt Template

Chinese:
你是一位精通汉语的大师。请根据下面给出的汉字，写出它所对应的标准普通话拼音，以json格式输出答案。示例如下：

```
<content>张</content>
输出：{
    "answer": "zhāng"
}
<content>我</content>
输出：{
    "answer": "wǒ"
}
<content>聂</content>
输出：{
    "answer": "niè"
}
```

本题目给出的汉字如下：
```
<content>{content}</content>
```

English Translation:
You are a master of Chinese. Please provide the standard Mandarin pinyin for the Chinese characters given below, and output the answer in JSON format. Example:

```
<content>张</content>
Output: {
    "answer": "zhāng"
}
<content>我</content>
Output: {
    "answer": "wǒ"
}
<content>聂</content>
Output: {
    "answer": "niè"
}
```

The Chinese character given in this problem is as follows:
```
<content>{content}</content>
```

Figure 16: Prompt template for the character-to-pinyin verification experiment (§ 3.3). `{content}` is replaced with the actual character.

Prompt Template

Chinese:
你是一位精通汉语的大师。请根据下面给出的一个“短语”，猜出和它在读音上最相近的四字成语（不要考虑语义），以json格式输出答案。示例如下：

```
<content>孰语盆笔</content>
输出：{
    "answer": "无与伦比"
}
<content>律师捕霜</content>
输出：{
    "answer": "屡试不爽"
}
<content>王杨捕劳</content>
输出：{
    "answer": "亡羊补牢"
}
```

本题目给出的短语如下：

```
<content>{content}</content>
```

English Translation:
You are a master of Chinese. Based on the given “phrase,” guess a four-character idiom whose pronunciation is closest to it (ignore meaning), and output the answer in JSON format. Example:

```
<content>孰语盆笔</content>
输出：{
    "answer": "无与伦比"
}
<content>律师捕霜</content>
输出：{
    "answer": "屡试不爽"
}
<content>王杨捕劳</content>
输出：{
    "answer": "亡羊补牢"
}
```

The phrase given in this problem is as follows:

```
<content>{content}</content>
```

Figure 17: Prompt template for the Homophone Recall task (§ 4.2.1). `{content}` is replaced with the actual homophonic “phrase”.

Prompt Template

Chinese:
你是一位精通汉语的大师。请从下面给出的包含谐音梗的句子中，抽取出关键的谐音部分，并写出它所对应的常用语（包括词语、成语），以json格式输出答案。示例如下：

<sentence>红色的圆形印章，简单来说，就是，朱圆章。</sentence>
输出：{
    "pun word": "朱圆章",
    "alternative word": "朱元璋"
}
<sentence>龙下凡，我接他，他说需要用雨水才能接他，于是我盛雨接龙。</sentence>
输出：{
    "pun word": "盛雨接龙",
    "alternative word": "成语接龙"
}
<sentence>据理力争的女律师被评为当代据理夫人。</sentence>
输出：{
    "pun word": "据理夫人",
    "alternative word": "居里夫人"
}

本题目给出的句子如下：
<sentence>{sentence}</sentence>

English Translation:
You are a master of Chinese. Please extract the key homophonic part from the given sentence that contains a pun, and provide the corresponding common word or idiom. Output the answer in JSON format. Example:

<sentence>红色的圆形印章，简单来说，就是，朱圆章。</sentence>
输出：{
    "pun word": "朱圆章",
    "alternative word": "朱元璋"
}
<sentence>龙下凡，我接他，他说需要用雨水才能接他，于是我盛雨接龙。</sentence>
输出：{
    "pun word": "盛雨接龙",
    "alternative word": "成语接龙"
}
<sentence>据理力争的女律师被评为当代据理夫人。</sentence>
输出：{
    "pun word": "据理夫人",
    "alternative word": "居里夫人"
}

The sentence given in this problem is as follows:
<sentence>{sentence}</sentence>

Figure 18: Prompt template for the Contextual Homophone Recognition task (§ 4.2.2). {sentence} is replaced with the actual pun sentence.

Prompt Template

Chinese:
你是一位精通汉语的大师。请根据下面给出的拼音，写出5个发音和该拼音完全相同的汉字，以json格式输出答案。示例如下：

```
<pinyin>chén</pinyin>
输出：{
    "answer": ["晨", "辰", "沉", "尘", "陈"]
}
<pinyin>jīng</pinyin>
输出：{
    "answer": ["茎", "鲸", "荆", "晶", "兢"]
}
<pinyin>yì</pinyin>
输出：{
    "answer": ["艺", "抑", "译", "忆", "易"]
}
```

本题目给出的拼音如下：

```
<pinyin>{content}</pinyin>
```

English Translation:
You are a master of Chinese. Please provide 5 Chinese characters that share exactly the same pronunciation as the given pinyin, and output the answer in JSON format. Example:

```
<pinyin>chén</pinyin>
Output: {
    "answer": ["晨", "辰", "沉", "尘", "陈"]
}
<pinyin>jīng</pinyin>
Output: {
    "answer": ["茎", "鲸", "荆", "晶", "兢"]
}
<pinyin>yì</pinyin>
Output: {
    "answer": ["艺", "抑", "译", "忆", "易"]
}
```

The pinyin given in this problem is as follows:

```
<pinyin>{content}</pinyin>
```

Figure 19: Prompt template for the pinyin2char test in RQ 1 (§ E, for the Character-Level Homophone Recall task). `{content}` is replaced with the actual pinyin.

Prompt Template

Chinese:

你是一位精通汉语的大师。请根据下面给出的汉字，写出5个发音和它完全相同的汉字，以json格式输出答案。示例如下：

```
<content>臣</content>
输出：{
    "answer": ["晨", "辰", "沉", "尘", "陈"]
}
<content>京</content>
输出：{
    "answer": ["茎", "鲸", "荆", "晶", "兢"]
}
<content>义</content>
输出：{
    "answer": ["艺", "抑", "译", "忆", "易"]
}
```

本题目给出的汉字如下：

```
<content>{content}</content>
```

English Translation:

You are a master of Chinese. Please provide 5 Chinese characters that share exactly the same pronunciation as the given character, and output the answer in JSON format. Example:

```
<content>臣</content>
Output: {
    "answer": ["晨", "辰", "沉", "尘", "陈"]
}
<content>京</content>
Output: {
    "answer": ["茎", "鲸", "荆", "晶", "兢"]
}
<content>义</content>
Output: {
    "answer": ["艺", "抑", "译", "忆", "易"]
}
```

The character given in this problem is as follows:

```
<content>{content}</content>
```

Figure 20: Prompt template for the char2pinyin test in RQ 1 (§ E, for the Character-Level Homophone Recall task). `{content}` is replaced with the actual character.

Prompt Template

Chinese:
你是一位精通汉语的大师。请根据下面给出的一个“短语”，写出它所对应的标准普通话拼音(各汉字对应的拼音之间以空格隔开），以json格式输出答案。示例如下：

<content>吴雨仑笔</content>
输出：{
    "answer": "wú yǔ lún bǐ"
}
<content>履弑布爽</content>
输出：{
    "answer": "lǚ shì bù shuǎng"
}
<content>哆咄易鳝</content>
输出：{
    "answer": "duō duō yì shàn"
}
本题目给出的短语如下：
<content>{content}</content>

English Translation:
You are a master of Chinese. Please provide the standard Mandarin pinyin for the given “phrase,” with each Chinese character’s pinyin separated by a space, and output the answer in JSON format. Example:

<content>吴雨仑笔</content>
Output：{
    "answer": "wú yǔ lún bǐ"
}
<content>履弑布爽</content>
Output：{
    "answer": "lǚ shì bù shuǎng"
}
<content>哆咄易鳝</content>
Output：{
    "answer": "duō duō yì shàn"
}
The phrase given in this problem is as follows:
<content>{content}</content>

Figure 21: Prompt template for the word2pinyin test in RQ 1 (§ E, for the Homophone Recall task). {content} is replaced with the actual homophonic “phrase”.

Prompt Template

Chinese:

```
你是一位精通汉语的大师。请根据下面给出的拼音，写出它所对应的成语，以json格式输出答案。示例如下：
<pinyin>wú yǔ lún bǐ</pinyin>
输出：{
    "answer": "无与伦比"
}
<pinyin>lǚ shì bù shuǎng</pinyin>
输出：{
    "answer": "屡试不爽"
}
<pinyin>duō duō yì shàn</pinyin>
输出：{
    "answer": "多多益善"
}
本题目给出的拼音如下：
<pinyin>{pinyin}</pinyin>
```

English Translation:

You are a master of Chinese. Please guess a four-character idiom whose pronunciation is closest to the given pinyin, and output the answer in JSON format. Example:

```
<pinyin>wú yǔ lún bǐ</pinyin>
Output: {
    "answer": "无与伦比"
}
<pinyin>lǚ shì bù shuǎng</pinyin>
Output: {
    "answer": "屡试不爽"
}
<pinyin>duō duō yì shàn</pinyin>
Output: {
    "answer": "多多益善"
}
```

The pinyin given in this problem is as follows:

```
<pinyin>{pinyin}</pinyin>
```

Figure 22: Prompt template for the pinyin2word test in RQ 1 (§ E, for the Homophone Recall task). `{pinyin}` is replaced with the actual pinyin.

Prompt Template

Chinese:
你是一位精通汉语的大师。请根据下面给出的句子以及从句子中提取出来的短语，写出短语在句子中的语境下所对应的标准普通话拼音(各汉字对应的拼音之间以空格隔开），以json格式输出答案。示例如下：

<sentence>大户人家的小王犯了两项家规，父亲决定对小王数罪并罚，也就是家法结合律。</sentence>
<content>家法结合律</content>
输出：{
    "answer": "jiā fǎ jié hé lǜ"
}
<sentence>有人踩了笛卡尔一脚，笛卡尔：没事儿，因为我是恕学家。</sentence>
<content>恕学家</content>
输出：{
    "answer": "shù xué jiā"
}
<sentence>食材越多，越容易作出更多的菜，或许这就是多多益膳。</sentence>
<content>多多益膳</content>
输出：{
    "answer": "duō duō yì shàn"
}
本题目给出的句子和短语如下：
<sentence>{sentence}</sentence>
<content>{content}</content>

English Translation:
You are a master of Chinese. Please provide the standard Mandarin pinyin for the given phrase extracted from the sentence, in the context of that sentence, with each Chinese character's pinyin separated by a space, and output the answer in JSON format. Example:

<sentence>大户人家的小王犯了两项家规，父亲决定对小王数罪并罚，也就是家法结合律。</sentence>
<content>家法结合律</content>
输出：{
    "answer": "jiā fǎ jié hé lǜ"
}
<sentence>有人踩了笛卡尔一脚，笛卡尔：没事儿，因为我是恕学家。</sentence>
<content>恕学家</content>
输出：{
    "answer": "shù xué jiā"
}
<sentence>食材越多，越容易作出更多的菜，或许这就是多多益膳。</sentence>
<content>多多益膳</content>
输出：{
    "answer": "duō duō yì shàn"
}

The sentence and phrase given in this problem are as follows:
<sentence>{sentence}</sentence>
<content>{content}</content>

Figure 23: Prompt template for the word2pinyin test in RQ 1 (§ E, for the Contextual Homophone Recognition task). {sentence} is replaced with the actual context sentence, and {content} with the actual word.

Prompt Template

Chinese:
你是一位精通汉语的大师。请根据下面给出的拼音，写出它所对应的最常用的短语（包括词语、成语），以json格式输出答案。示例如下：

```
<pinyin>jiā fǎ jié hé lǜ</pinyin>
输出：{
    "answer": "加法结合律"
}
<pinyin>shù xué jiā</pinyin>
输出：{
    "answer": "数学家"
}
<pinyin>duō duō yì shàn</pinyin>
输出：{
    "answer": "多多益善"
}
```

本题目给出的拼音如下：

```
<pinyin>{pinyin}</pinyin>
```

English Translation:
You are a master of Chinese. Please provide the most commonly used phrase (including words or chengyu) corresponding to the given pinyin, and output the answer in JSON format. Example:

```
<pinyin>jiā fǎ jié hé lǜ</pinyin>
Output：{
    "answer": "加法结合律"
}
<pinyin>shù xué jiā</pinyin>
Output：{
    "answer": "数学家"
}
<pinyin>duō duō yì shàn</pinyin>
Output：{
    "answer": "多多益善"
}
```

The pinyin given in this problem is as follows:

```
<pinyin>{pinyin}</pinyin>
```

Figure 24: Prompt template for the pinyin2word test in RQ 1 (§ E, for the Contextual Homophone Recognition task). {pinyin} is replaced with the actual pinyin.

Prompt Template

Chinese:
你是一位使用汉语的作词人，需要根据既有的参考短句，创作出一句和它押韵的短句（且韵脚字不雷同）。注意，如果两个汉字的韵母出现在下面的同一组中，我们认为它们相互押韵（不要求声调相同）。

<groups>
第一组：a、ia、ua（如“他”“家”“瓜”）；
第二组：o、uo（如“波”“诺”）；
第三组：e（如“鹅”）；
第四组：ie、üe（如“接”“月”）；
第五组：er（如“而”）；
第六组：i、ü（如“知”“及”“鱼”）；
第七组：u（如“姑”）；
第八组：ai、uai（如“来”“乖”）；
第九组：ei、uei（如“雷”“鬼”）；
第十组：ao、iao（如“老”“交”）；
第十一组：ou、iou（如“楼”“有”）；
第十二组：an、uan、ian、üan（如“安”“关”“先”“远”）；
第十三组：en、uen（如“恩”“文”）；
第十四组：in、ün（如“音”“云”）；
第十五组：ang、uang、iang（如“昂”“光”“想”）；
第十六组：ing（如“英”）；
第十七组：eng、ueng（如"成"“风”“翁”）；
第十八组：ong、iong（如“龙”“雄”）。
</groups>

请以json格式输出答案。示例如下：

<reference>一瞬间想起了那时候的自己</reference>
输出：{
"answer": "在塞纳河边的回忆你和谁在一起"
}
<reference>我有了你生命才有希望</reference>
输出：{
"answer": "破碎的心失去依傍"
}
<reference>付出所有的青春不留遗憾</reference>
输出：{
"answer": "你知道的我有半夜敲文字的习惯"
}

请根据下面的参考短句生成押韵的短句：
<content>{content}</content>

(continue on next page)

(continue from previous page)

English Translation:

You are a lyricist writing in Chinese. Based on the given reference sentence, create a new sentence that rhymes with it (the final character should not be the same as the reference's). Note: If the finals (yunmu) of two Chinese characters appear in the same group below, they are considered to rhyme (tone does not need to match).

```
<groups>
Group 1: a、ia、ua (e.g., "他", "家", "瓜");
Group 2: o、uo (e.g., "波", "诺");
Group 3: e (e.g., "鹅");
Group 4: ie、üe (e.g., "接", "月");
Group 5: er (e.g., "而");
Group 6: i、ü (e.g., "知", "及", "鱼");
Group 7: u (e.g., "姑");
Group 8: ai、uai (e.g., "来", "乖");
Group 9: ei、uei (e.g., "雷", "鬼");
Group 10: ao、iao (e.g., "老", "交");
Group 11: ou、iou (e.g., "楼", "有");
Group 12: an、uan、ian、üan (e.g., "安", "关", "先", "远");
Group 13: en、uen (e.g., "恩", "文");
Group 14: in、ün (e.g., "音", "云");
Group 15: ang、uang、iang (e.g., "昂", "光", "想");
Group 16: ing (e.g., "英");
Group 17: eng、ueng (e.g., "成", "风", "翁");
Group 18: ong、iong (e.g., "龙", "雄").
</groups>
```

Please output the answer in JSON format. Example:

```
<reference>一瞬间想起了那时候的自己</reference>
Output: {
"answer": "在塞纳河边的回忆你和谁在一起"
}
<reference>我有了你生命才有希望</reference>
Output: {
"answer": "破碎的心失去依傍"
}
<reference>付出所有的青春不留遗憾</reference>
Output: {
"answer": "你知道的我有半夜敲文字的习惯"
}
```

Please generate a sentence based rhyming with the reference sentence given below:

```
<reference>{content}</reference>
```

Figure 25: Prompt template for the Rhyming Sentence Generation task (1-rhyme setting, § 4.3). `{content}` is replaced with the actual reference sentence.

Prompt Template

Chinese:
你是一位使用汉语的作词人，需要根据既有的参考短句，创作出一句和它双押的短句，要求它最后两个字和参考短句的最后两个字分别押韵（且韵脚字不雷同）。注意，如果两个汉字的韵母出现在下面的同一组中，我们认为它们相互押韵（不要求声调相同）。

<groups>
第一组：a、ia、ua（如“他”“家”“瓜”）；
第二组：o、uo（如“波”“诺”）；
第三组：e（如“鹅”）；
第四组：ie、üe（如“接”“月”）；
第五组：er（如“而”）；
第六组：i、ü（如“知”“及”“鱼”）；
第七组：u（如“姑”）；
第八组：ai、uai（如“来”“乖”）；
第九组：ei、uei（如“雷”“鬼”）；
第十组：ao、iao（如“老”“交”）；
第十一组：ou、iou（如“楼”“有”）；
第十二组：an、uan、ian、üan（如“安”“关”“先”“远”）；
第十三组：en、uen（如“恩”“文”）；
第十四组：in、ün（如“音”“云”）；
第十五组：ang、uang、iang（如“昂”“光”“想”）；
第十六组：ing（如“英”）；
第十七组：eng、ueng（如"成"“风”“翁”）；
第十八组：ong、iong（如“龙”“雄”）。
</groups>

请以json格式输出答案。示例如下：

<reference>一瞬间想起了那时候的自己</reference>
输出：{
"answer": "孤独的我在梦里寻觅"
}
<reference>我有了你生命才有希望</reference>
输出：{
"answer": "多希望有一个让我依靠的肩膀"
}
<reference>付出所有的青春不留遗憾</reference>
输出：{
"answer": "用慈爱做花环，人与人不冷淡"
}
请根据下面的参考短句生成押韵的短句：
<content>{content}</content>

(continue on next page)

(continue from previous page)

English Translation:
You are a lyricist writing in Chinese. Based on the given reference sentence, create a new double-rhymed sentence, such that the last two characters of your sentence rhyme with the last two characters of the reference sentence, respectively (the last two characters should not be the same as the reference's, too). Note: If the finals (yunmu) of two Chinese characters appear in the same group below, they are considered to rhyme (tone does not need to match).

```
<groups>
Group 1: a、ia、ua (e.g., "他", "家", "瓜");
Group 2: o、uo (e.g., "波", "诺");
Group 3: e (e.g., "鹅");
Group 4: ie、üe (e.g., "接", "月");
Group 5: er (e.g., "而");
Group 6: i、ü (e.g., "知", "及", "鱼");
Group 7: u (e.g., "姑");
Group 8: ai、uai (e.g., "来", "乖");
Group 9: ei、uei (e.g., "雷", "鬼");
Group 10: ao、iao (e.g., "老", "交");
Group 11: ou、iou (e.g., "楼", "有");
Group 12: an、uan、ian、üan (e.g., "安", "关", "先", "远");
Group 13: en、uen (e.g., "恩", "文");
Group 14: in、ün (e.g., "音", "云");
Group 15: ang、uang、iang (e.g., "昂", "光", "想");
Group 16: ing (e.g., "英");
Group 17: eng、ueng (e.g., "成", "风", "翁");
Group 18: ong、iong (e.g., "龙", "雄").
</groups>
```

Please output the answer in JSON format. Example:

```
<reference>一瞬间想起了那时候的自己</reference>
输出: {
"answer": "在塞纳河边的回忆你和谁在一起"
}
<reference>我有了你生命才有希望</reference>
输出: {
"answer": "破碎的心失去依傍"
}
<reference>付出所有的青春不留遗憾</reference>
输出: {
"answer": "你知道的我有半夜敲文字的习惯"
}
```

Please generate a sentence based double-rhyming with the reference sentence given below:
`<reference>{content}</reference>`

Figure 26: Prompt template for the Rhyming Sentence Generation task (2-rhyme setting, § 4.3). `{content}` is replaced with the actual reference sentence.

Prompt Template

Chinese:
你是一位精通汉语的大师。请从四个选项中，选出在读音上和给出的词语最接近的一项（不要考虑语义），以json格式输出答案。示例如下：

<content>技法</content>
<choices>
A. 家法
B. 依法
C. 办法
D. 立法
</choices>
输出：{
    "answer": "D"
}
<content>增添</content>
<choices>
A. 今天
B. 春天
C. 昨天
D. 成天
</choices>
输出：{
    "answer": "B"
}
<content>渐渐</content>
<choices>
A. 健全
B. 建议
C. 间接
D. 见面
</choices>
输出：{
    "answer": "D"
}
本题目给出的词语和选项如下：
<content>{content}</content>
<choices>
A. {A}
B. {B}
C. {C}
D. {D}
</choices>

(continue in the next page)

(continue from the previous page)

English Translation:
You are a master of Chinese. From the four options, choose the one whose pronunciation is closest to the given word (ignore meaning), and output the answer in JSON format. Example:

```
<content>技法</content>
<choices>
A. 家法
B. 依法
C. 办法
D. 立法
</choices>
Output: {
    "answer": "D"
}
<content>增添</content>
<choices>
A. 今天
B. 春天
C. 昨天
D. 成天
</choices>
Output: {
    "answer": "B"
}
<content>渐渐</content>
<choices>
A. 健全
B. 建议
C. 间接
D. 见面
</choices>
Output: {
    "answer": "D"
}
```

The word and choices given in this problem are as follows:

```
<content>{content}</content>
<choices>
A. {A}
B. {B}
C. {C}
D. {D}
</choices>
```

Figure 27: Prompt template for the Similarity Comparison task (in word form, § 4.4). `{content}` is replaced with the actual reference word, and `{A-D}` with the actual answer choices.

Prompt Template

Chinese:

你是一位精通汉语的大师。请从四个拼音选项中，选出和给出的拼音读音最接近的一项，以json格式输出答案。示例如下：

```
<content>jì fǎ</content>
<choices>
A. jiā fǎ
B. yī fǎ
C. bàn fǎ
D. lì fǎ
</choices>
输出：{
    "answer": "D"
}
<content>zēng tiān</content>
<choices>
A. jīn tiān
B. chūn tiān
C. zuó tiān
D. chéng tiān
</choices>
输出：{
    "answer": "B"
}
<content>jiàn jiàn</content>
<choices>
A. jiàn quán
B. jiàn yì
C. jiàn jiē
D. jiàn miàn
</choices>
输出：{
    "answer": "D"
}
本题目给出的拼音和选项如下：
<content>{content}</content>
<choices>
A. {A}
B. {B}
C. {C}
D. {D}
</choices>
```

(continue in the next page)

(continue from the previous page)

English Translation:
You are a master of Chinese. From the four pinyin options, choose the one whose pronunciation is closest to the given one, and output the answer in JSON format. Example:

```
<content>jì fǎ</content>
<choices>
A. jiā fǎ
B. yī fǎ
C. bàn fǎ
D. lì fǎ
</choices>
Output: {
    "answer": "D"
}
<content>zēng tiān</content>
<choices>
A. jīn tiān
B. chūn tiān
C. zuó tiān
D. chéng tiān
</choices>
Output: {
    "answer": "B"
}
<content>jiàn jiàn</content>
<choices>
A. jiàn quán
B. jiàn yì
C. jiàn jiē
D. jiàn miàn
</choices>
Output: {
    "answer": "D"
}
```

The pinyin and choices given in this problem are as follows:

```
<content>{content}</content>
<choices>
A. {A}
B. {B}
C. {C}
D. {D}
</choices>
```

Figure 28: Prompt template for the Similarity Comparison task (in pinyin form, § 4.4). `{content}` is replaced with the actual reference pinyin, and `{A-D}` with the actual answer choices.

Speech Template

Chinese:
你是一位精通汉语的大师。请听写出下面我读出的四字成语，以json格式输出答案。本题给出的成语是:{content}。

English Translation:
You are a master of Chinese. Please transcribe the following four-character chengyu I read aloud and output the answer in JSON format. The given chengyu is: {content}.

Figure 29: Speech template used for text-to-speech in the audio-model evaluation (Homophone Recall). {content} is replaced with the actual chengyu.

Prompt Template

Chinese:
请回答音频中的问题。
{audio_file}

English Translation:
Please answer the question in the audio.
{audio_file}

Figure 30: Prompt template for in the audio-model experiment (§ 5.4.3), paired with the audio file generated via text-to-speech. {audio_file} is replaced with the actual audio file.

Human Instruction

Chinese:
任务：
写出对应汉字的普通话拼音。声调用数字附在最后（即第几声）。如：
问题：一
回答：yi1

不会的可以猜一个。
闭卷！

English Translation:
Write the Mandarin pinyin corresponding to each given Chinese character. Indicate the tone by appending a number at the end (representing which tone it is). For example:
Question: 一
Answer: yi1

If you're unsure, you may make a guess.
Closed-book!

Figure 31: Instruction for the character-to-pinyin verification experiment (§ 3.3).

Human Instruction

Chinese:
任务:
猜一个和给出的"短语"读音上最接近的成语。
由于拼音输入法本身就可能泄露答案，所以应当先猜测完再输入。（不能根据输入法的提示修改）。
如果猜不到，则在空格中填一个句号。

English Translation:
Task:
Guess the idiom whose pronunciation is closest to the given "phrase."
Since pinyin input methods may inadvertently reveal the answer, you should make your guess before typing it (do not rely on input method suggestions).
If you cannot guess, write a period (.) in the blank.

Figure 32: Instruction for the Homophone Recall task (§ 4.2.1).

Human Instruction

Chinese:
任务:
从这些谐音梗句子中，提取它们的谐音部分，以及谐音部分所对应的常用语。如:
句子：大户人家的小王犯了两项家规，父亲决定对小王数罪并罚，也就是家法结合律。
谐音部分：家法结合律
对应常用语：加法结合律

注意，选取的时候尽可能地往小范围取，而非往大范围取。比如上面的例子中，谐音部分、对应常用语可以分别只写写"家法"和"加法"。

English Translation:
Task:
From these homophonic pun sentences, extract the homophonic part and the common phrase it corresponds to. For example:
Sentence: 大户人家的小王犯了两项家规，父亲决定对小王数罪并罚，也就是家法结合律。
Homophonic part: 家法结合律
Corresponding common phrase: 加法结合律

Note: When selecting the phrase, try to keep the scope as narrow as possible rather than overly broad. For instance, in the example above, the homophonic part and the corresponding phrase could simply be "家法" and "加法", respectively.

Figure 33: Instruction for the Contextual Homophone Recognition task (§ 4.2.2).

Human Instruction

Chinese:
任务:
从4个选项中，选择你感觉的和参考词读音最接近的一项（只考虑读音，不考虑语义）。
如果不确定，请蒙一个。

English Translation:
Task:
From the four options, choose the one whose pronunciation is closest to the reference word (consider sound only, not meaning).
If you are unsure, make a guess.

Figure 34: Instruction for the Similarity Comparison task (in word form, § 4.4).

Human Instruction

Chinese:
任务:
从4个选项中，选择你感觉的和参考词读音最接近的一项。
如果不确定，请蒙一个。

English Translation:
Task:
From the four options, choose the one whose pronunciation you think is closest to the reference word.
If you are unsure, make a guess.

Figure 35: Instruction for the Similarity Comparison task (in pinyin form, § 4.4).

	A	B	C	D	E	F	G	H	I	J	K	L
1	id	参考词	选项A	选项B	选项C	选项D	回答					
2	2	前身	潜能	乾坤	前世	前期						
3	8	宣誓	优势	形势	监事	电视						
4	80	人治	肉质	惩治	材质	增至						
5	245	天时	天子	天使	天然	天上						
6	499	盲人	名城	王牌	上层	残忍						
7	378	砂糖	终极	相连	出名	家常						
8	291	联席	联盟	联谊	联想	联系						
9	411	基建	之间	空间	阶段	期间						
10	52	如实	肉食	同时	务实	熟食						
11	250	上场	商场	当场	在场	收场						
12	275	新任	欣然	新闻	新人	心脏						
13	286	出世	出水	出去	初始	出资						
14	125	大多	大道	大都	大作	大家						
15	418	半路	贯彻	栏目	步入	到处						
16	114	洞察	调查	红茶	绿茶	督察						
17	224	收受	收拾	收入	收手	收口						
18	260	油性	游行	游客	邮箱	油品						
19	40	基建	基金	机电	基点	机会						
20	23	对劲	对接	对应	对象	对心						

任务:
从4个选项中，选择你感觉的和参考词读音最接近的一项（只考虑读音，不考虑语义）。
如果不确定，请蒙一个。

Figure 36: Screenshot of the Similarity Comparison .xlsx file provided to human participants.