

MTGBench: A Multi-Turn Game Benchmark for Diverse LLM Capability Assessment

Alisa Alenicheva¹ Timur Zykov^{1,2} Artsiom Saladukha¹ Vladislav Negodin²

¹Yandex School of Data Analysis ²Yandex

Abstract

Existing LLM reasoning benchmarks are largely static, single-turn, English-centric, and reliant on biased LLM-as-a-judge scoring. We present **MTGBench**, a bilingual (English/Russian) suite of **seven** multi-turn puzzle games—Wordle, Word Chain, Situation Puzzles, 20 Questions, Wheel of Fortune, 20 Features, and Alias—spanning deductive, semantic, lateral, consistency, and knowledge-retrieval reasoning. Four games use algorithmic checkers; the remaining three use a constrained LLM-as-a-rule-enforcer validated against human annotators, and episodes are regenerated natively per language rather than translated. Across 16 frontier models and a human baseline comprising 476 sessions we find that (i) rankings are highly task-dependent, (ii) chain-of-thought is near-essential for letter-level games but uneven on knowledge-driven ones, and (iii) the strongest reasoning-enabled LLMs match or exceed our collected Russian human baseline across the entire suite. Game scores align most consistently with cognitive batteries probing Causal Reasoning, Pragmatics, Planning, and Common-sense Reasoning, though these associations are game-specific rather than a single executive-function signal. Code and data are available in [MTGBench github](#).

1 Introduction

Most existing LLM benchmarks—spanning instruction following (Zhou et al., 2023), factuality (Cheng et al., 2025), knowledge (Wang et al., 2024; Phan et al., 2026), mathematics (Hendrycks et al., 2021; Glazer et al., 2025), and code generation (Jimenez et al., 2024; Jain et al., 2024)—are single-turn. They provide all context upfront, leaving untested whether a model can adapt to new information, maintain consistency, or plan across multiple steps (Laban et al., 2025). Scaling multi-turn evaluation remains difficult: open-ended interactions require expensive human labor or rely on

subjective, inherently biased LLM-as-a-judge scoring (Zheng et al., 2023; Bai et al., 2024; Wataoka et al., 2025; Thakur et al., 2025). Furthermore, English remains overrepresented in multilingual evaluation, and locally curated benchmarks align better with human judgments than translated ones (Wu et al., 2025); reasoning performance also remains sensitive to language (Qi et al., 2025; Yong et al., 2025).

Puzzle games offer a robust, controlled alternative. Constrained environments admit fully algorithmic checkers (Li et al., 2026; Gui et al., 2025) or tightly scoped *LLM-as-a-rule-enforcer* paradigms that verify objective game states rather than subjective output quality. Because game logic is separated from linguistic content, episodes can be generated natively in each language, avoiding direct translation bias. Games also yield reliable, aggregable scores (Chalamalasetti et al., 2023; Badola et al., 2025). Prior experiments further suggest that interactive games provide evaluation signals complementary to static benchmarks for cognitive functions (Momentè et al., 2025).

We present **MTGBench**, a **bilingual** (English/Russian) benchmark of **seven** multi-turn puzzle games testing diverse reasoning skills: **Wordle** (deductive elimination), **Word Chain** (semantic chaining), **Situation Puzzles** (lateral thinking), **Wheel of Fortune** (letter-by-letter guessing), **20 Questions** (logical consistency), **20 Features** (incremental deduction), and **Alias** (strategic entity identification). Four games use algorithmic checkers, while three employ a constrained, human-validated LLM rule-enforcer.

Evaluating 16 frontier models (Qwen, DeepSeek, GPT, Gemini) across Answer-Only (AO) and Chain-of-Thought (CoT)/Reasoning regimes—alongside a 476-session human baseline on identical task instances—reveals three key insights: (i) model rankings are highly *task-dependent*; (ii) CoT is near-essential for letter-level games but

yields only modest improvements in knowledge-retrieval games; and (iii) the human baseline outperforms AO models on letter-based and rule-following games, whereas the strongest reasoning-enabled LLMs match or exceed it on all seven games. Finally, across the 12 models included in the AO correlation analysis, MTGBench performance is most consistently associated with benchmarks of causal reasoning, pragmatics, planning, and common-sense reasoning.

Our main contributions are:

1. A bilingual multi-turn benchmark of seven games combining algorithmic checkers and a human-validated LLM rule-enforcer, powered by a language-agnostic generation pipeline.
2. A comprehensive evaluation of 16 frontier models and a human baseline, demonstrating that multi-turn performance is task-dependent and CoT benefits are non-uniform.
3. A model-level correlation analysis against 11 cognitive benchmarks, revealing heterogeneous, game-specific association patterns rather than a single executive-function signal.
4. A full release of the benchmark code and data at <https://github.com/skyberg11/MTGBench>.

2 Background

Game-based evaluation reframes LLM testing as agentic play within a constrained environment. It offers controlled, goal-directed settings in which language use is assessed through interaction, specific model capabilities can be targeted, and performance can be quantified by task success (Schlangen, 2023). For LLM benchmarks, games can provide explicit state-based scoring or tightly constrained “LLM-as-a-rule-enforcer” judges. Multi-turn games require models to integrate feedback across turns—revealing planning and working-memory failures missed by single-turn QA tasks. Finally, game-based benchmark episodes can be generated natively in different languages using local lexical resources, reducing reliance on translation and its associated biases.

Prior work spans three overlapping directions:

1. **RL Environments:** Frameworks like TextWorld (Côté et al., 2019) provide text-adventure sandboxes, but primarily target

reinforcement learning rather than LLM probing. LMRL Gym (Abdulhai et al., 2023) extends this paradigm to language-model policies specifically, providing offline datasets and interactive simulators for eight multi-turn tasks and comparing few-shot prompting baseline with offline and online RL fine-tuning, rather than primarily benchmarking pretrained LLMs at inference time.

2. **Chat as a Game Board:** *clembench* (Chalamasetti et al., 2023) evaluates five English interaction settings. LMRL Gym also includes three interactive-dialogue environments — 20 Questions, Guess My City, and Car Dealer — while Paprika (Tajwar et al., 2025) uses 20 Questions and related interactive tasks to train and assess transferable information-gathering behavior. The classic 20 Questions format underlies our Alias game.
3. **Deterministic Puzzles:** LogicGame (Gui et al., 2025) provides English–Chinese rule-based reasoning problems, but evaluates each problem with a single prompt rather than through interaction with an environment. Conversely, Multi-Turn Puzzles (MTP) (Badola et al., 2025) offers a multi-turn suite but relies on simplified constraints. Specific methodological differences between MTGBench and MTP are detailed in Section 3.

Competitive and live game arenas form a related but methodologically distinct line. GAMA-Bench evaluates decision making in eight multi-agent game-theoretic scenarios, and ReTA includes an Online Racing setting in which language-model agents compete directly (Huang et al., 2024; Duan et al., 2024). TextArena offers more than 57 single-, two-, and multi-player text environments and an online system that rates submitted models from games against humans and other models (Guertler et al., 2025). GameArena instead collects model–human interactions in three live computer games designed to probe deductive and inductive reasoning (Hu et al., 2025). In these arena-style settings, systems are compared through model–model competition or a shared pool of human/opponent interactions; comparable measurements therefore require interactions against the same counterpart distribution. MTGBench instead uses pointwise evaluation on matched, fixed instances: each model’s score is computed independently, so adding a model does

not require replaying pairwise matches or changing existing scores.

MTGBench targets multi-turn puzzle gameplay, independent per-model evaluation, and a mixed judging methodology for both formal and open-form answers. It uses algorithmic checkers for four games and a constrained LLM rule-enforcer for the remaining three. Table 1 compares independently runnable game and puzzle benchmarks along these axes.

3 MTGBench

Our benchmark features seven multi-turn games evaluating deductive reasoning, logical consistency, strategic planning, and information retrieval. Several games adapt established formats from prior work (Chalamalasetti et al., 2023; Abdulhai et al., 2023; Tajwar et al., 2025; Badola et al., 2025); TextArena and GameArena provide related examples of interactive guessing and game-based evaluation (Guertler et al., 2025; Hu et al., 2025). We instantiate these formats in a fixed bilingual evaluation protocol. Our contribution is the combination of these tasks with matched English and Russian instances, mixed algorithmic and constrained-judge evaluation, and a common scoring framework. A comparison with prior frameworks is provided in Table 1.

Wordle requires guessing a five-letter word from color-coded feedback. Unlike the Multi-Turn Puzzles (MTP) benchmark (Badola et al., 2025), which evaluates each game against one of ten 40-word vocabularies, ours does not provide the model with a candidate list. Furthermore, unlike the English-only Wordle setup in *clembench*, which uses fixed target and valid-guess lists (Chalamalasetti et al., 2023), we enforce strict dictionary and part-of-speech validation (nouns only) using lexical resources (NLTK’s interface to Princeton WordNet (Steven Bird, 2009; Fellbaum, 1998) for English; RuWordNet (Loukachevitch et al., 2016) for Russian). Invalid guesses trigger validation errors and consume one benchmark turn, requiring the model to rely on its intrinsic linguistic knowledge rather than a prompt-provided candidate set.

Word Chain asks players to sequentially name cities starting with the last letter of the previous one. Unlike MTP, which constructs a 500-word lexicon sampled from SCOWL for each game (Badola et al., 2025), our task operates over cities and does not provide the model with an

explicit allowed-word list; moves are validated against a fixed snapshot of the Geoapify World Cities Database, an OpenStreetMap-derived locality database (Geoapify, 2026).

20 Questions (Consistency) tests if a model can answer yes/no questions about a secret word without self-contradiction. Unlike MTP, where the model selects a secret noun from a predefined WordNet lexicon and answers questions about that noun and its hypernyms (Badola et al., 2025), we reveal a synthetic target and its hierarchical attributes to the model. This design is intended to reduce dependence on world knowledge and target recall and to emphasize answer consistency (Appendix A.4).

The remaining four games extend familiar guessing and puzzle formats within our bilingual protocol: **Situation Puzzles** requires asking yes/no questions to deduce the hidden explanation behind an ambiguous scenario. **Wheel of Fortune (WoF)** asks models to strategically guess letters of a partially revealed word (e.g., _ _ A _ _), testing letter-frequency knowledge. **20 Features** tests incremental deduction by revealing object properties one by one until the model identifies the item. **Alias** challenges the model to identify a famous person by asking strategic yes/no questions to efficiently narrow a vast candidate space. Formal definitions are provided in Appendix A.

Benchmark Design Principles. All seven games are bilingual (English and Russian); Wordle and Wheel of Fortune support built-in difficulty stratification (e.g., word rarity tiers). Four games rely on deterministic algorithmic judges. The remaining three open-ended games (Situation Puzzles, 20 Features, Alias) employ LLM-based judges with tightly constrained win/loss conditions; their agreement with independent reference judgments is reported in Section 4.4.

3.1 Experimental Setup

We evaluated 100 episodes per game across identical initial conditions using a fixed random seed (42). Games are divided into two scoring categories:

Speed games (Wordle, 20 Features, WoF, Situation Puzzles, and Alias) require guessing the target as quickly as possible. Scores equal 1 when the target is solved at or before an optimal, game-specific `min_steps` threshold and then decay linearly toward 0 as the maximum turn horizon is approached. Because the scoring formulas use the zero-based index of the terminal turn, a successful episode on

Benchmark	Multi-turn	# Tasks	Bilingual	Judge Type	Diff. Levels
LMRL Gym (Abdulhai et al., 2023)	✓	8		Algo + LLM	
Multi-Turn Puzzles (Badola et al., 2025)	✓	5		Algo	
<i>clembench</i> (Chalamalasetti et al., 2023)	✓	10		Algo	
TextWorld (Côté et al., 2019)	✓	1		Algo	✓
LogicGame (Gui et al., 2025)		23	✓	Algo	✓
MTGBench (Ours)	✓	7	✓	Algo + LLM	✓

Table 1: Comparison of game-based LLM evaluation benchmarks. “Judge Type” indicates whether evaluation is fully algorithmic (*Algo*), combines algorithmic checkers with a constrained LLM-as-rule-enforcer judge or LLM for environment feedback (*Algo + LLM*); or uses rule-based game outcomes together with live human interaction (*Algo + Human*). LMRL Gym is marked *Algo + LLM* because its dialogue tasks use LLM as environment simulators. “Diff. Levels” indicates built-in difficulty stratification.

the final allowed turn retains a small positive score, whereas a failed episode scores 0. For instance, the WoF optimum is defined relative to half the number of distinct letters in the word, while the minimum for Situation Puzzles is one turn. WoF episodes are sampled across three difficulty levels stratified by corpus letter frequency (Appendix A.5).

Survival games (20 Questions and Word Chain) test how long the model avoids making an incorrect move. Models score 1 for reaching the maximum turn limit (20 or 25, respectively), or receive a fractional score based on valid turns survived.

For games requiring an LLM judge (Alias, 20 Features, Situation Puzzles), we use DeepSeek V4 Flash with temperature 0 and reasoning disabled. To prevent data leakage, the judge prompts were tuned on a held-out dataset that does not overlap with evaluation data. Judge quality is detailed in Section 4.4.

4 Results

4.1 Evaluation Setup

We evaluated four model families (Qwen, DeepSeek, GPT, and Gemini) under two inference formats. **Answer Only (AO)** instructs the model to produce direct answers, explicitly disabling native reasoning (models lacking this control were excluded). **CoT/Reasoning** activates native reasoning (at medium or default effort) or applies standard chain-of-thought prompts for models without native reasoning.

To ensure fairness, all evaluations used default temperatures and a fixed random seed (42). The seed fixes the sampled game instances and their initial states across models; it does not make API generation deterministic. At the analysis cutoff, 43 of 39,200 scheduled model episodes (0.11%) were

incomplete due to openrouter issues and were excluded from all reported analyses (see Appendix 9 for details). Results are summarized in Table 2, with confidence intervals reported in Appendix G.

4.2 Key Findings

Our benchmark reveals several crucial patterns in multi-turn capabilities:

1. Diverse Capability Profiles: Model rankings vary across games, illustrating the additional information provided by a multifaceted benchmark over a single-task evaluation.

2. The Impact of Reasoning: CoT/Reasoning produces the largest gains in letter-level manipulation (Wordle, Wheel of Fortune). In Wordle, the non-reasoning Qwen configurations score between 0.01 and 0.21 in AO mode and substantially higher with CoT. Stronger models (GPT 5.4, DeepSeek V4 Pro) also improve when reasoning is enabled. On knowledge-intensive and pragmatic games (Situation Puzzles, Alias), the observed differences are generally smaller and vary by model.

3. Cross-Lingual Performance: English and Russian rank models almost identically, yet weaker models incur a consistent penalty on Russian letter-based games, plausibly linked to Cyrillic subword tokenization; we analyze both effects in Section 4.3.

4. Top Performer & Error Modes: Gemini 3.1 Pro is the strongest overall model, achieving near-perfect win rates on Wordle, Wheel of Fortune, Word Chain, and 20 Questions. Manual error analysis (Figure 9 and Tables 14–20) identifies two recurring inefficiencies in the inspected failures: asking overly narrow early questions instead of broad queries that reduce the search space, and occasionally ignoring accumulated dialogue history, leading to redundant or contradictory inquiries.

Model	Mode	English (EN)							Russian (RU)							Total
		Wdl	WC	SitP	20Q	WoF	20F	Als	Wdl	WC	SitP	20Q	WoF	20F	Als	
Qwen3 30B A3B Instruct 2507	AO	0.014	0.081	0.124	0.340	0.295	0.247	0.140	0.177	0.010	0.094	0.049	0.346	0.272	0.199	0.157
	CoT	0.697	0.219	0.537	0.485	0.693	0.248	0.214	0.442	0.294	0.246	0.261	0.533	0.454	0.235	0.321
Qwen3 30B A3B Thinking 2507	Reason	0.855	0.860	0.491	0.997	0.647	0.333	0.309	0.642	0.452	0.645	0.394	0.978	0.546	0.372	0.304
Qwen3 235B A22B Instruct 2507	AO	0.210	0.205	0.536	0.430	0.549	0.357	0.292	0.368	0.036	0.124	0.365	0.439	0.411	0.362	0.235
	CoT	0.819	0.488	0.616	0.643	0.746	0.377	0.263	0.565	0.515	0.323	0.454	0.563	0.664	0.494	0.295
Qwen3 235B A22B Thinking 2507	Reason	0.911	0.969	0.622	0.982	0.822	0.581	0.409	0.757	0.639	0.816	0.480	0.976	0.725	0.662	0.347
Qwen3.5 35B A3B	AO	0.107	0.005	0.265	0.337	0.521	0.343	0.234	0.259	0.014	0.048	0.157	0.263	0.301	0.309	0.190
	Reason	0.899	0.849	0.562	0.991	0.795	0.615	0.429	0.734	0.624	0.762	0.363	0.978	0.656	0.693	0.441
DeepSeek V3.1 Terminus	AO	0.636	0.343	0.337	0.690	0.673	0.439	0.436	0.508	0.052	0.289	0.341	0.676	0.463	0.464	0.336
	Reason	0.911	0.954	0.405	0.993	0.823	0.599	0.540	0.747	0.826	0.666	0.334	0.945	0.709	0.611	0.426
DeepSeek V4 Flash	AO	0.315	0.448	0.469	0.598	0.647	0.419	0.410	0.472	0.166	0.424	0.323	0.490	0.462	0.488	0.236
	Reason	0.891	0.534	0.529	0.995	0.836	0.548	0.417	0.679	0.762	0.534	0.389	0.885	0.705	0.611	0.393
DeepSeek V4 Pro	AO	0.585	0.678	0.465	0.884	0.719	0.483	0.422	0.605	0.341	0.421	0.231	0.657	0.619	0.636	0.330
	Reason	0.944	0.827	0.581	1.000	0.853	0.592	0.508	0.758	0.876	0.580	0.468	0.923	0.732	0.707	0.451
GPT OSS 120B	Reason	0.903	0.871	0.525	0.867	0.748	0.572	0.399	0.698	0.618	0.833	0.392	0.970	0.658	0.588	0.390
GPT 4.1	AO	0.414	0.506	0.476	0.608	0.659	0.542	0.452	0.523	0.141	0.386	0.359	0.608	0.508	0.652	0.449
	CoT	0.712	0.947	0.681	0.730	0.760	0.544	0.493	0.695	0.167	0.581	0.486	0.614	0.651	0.604	0.436
GPT 5.4 Nano	AO	0.052	0.044	0.209	0.263	0.344	0.223	0.112	0.178	0.000	0.033	0.152	0.204	0.216	0.306	0.086
	Reason	0.853	0.918	0.445	0.660	0.777	0.467	0.229	0.621	0.663	0.851	0.411	0.733	0.720	0.501	0.282
GPT 5.4 Mini	AO	0.147	0.462	0.399	0.463	0.574	0.444	0.243	0.390	0.060	0.287	0.299	0.368	0.338	0.468	0.203
	Reason	0.913	0.662	0.438	0.828	0.829	0.582	0.345	0.657	0.791	0.502	0.421	0.940	0.756	0.700	0.363
GPT 5.4	AO	0.611	0.636	0.479	0.977	0.755	0.589	0.320	0.624	0.343	0.494	0.382	0.906	0.570	0.601	0.310
	Reason	0.961	0.961	0.494	0.987	0.883	0.642	0.424	0.765	0.940	0.964	0.438	0.970	0.845	0.748	0.486
Gemini 3.1 Flash Lite	AO	0.355	0.578	0.345	0.967	0.744	0.471	0.450	0.559	0.106	0.401	0.306	0.906	0.550	0.522	0.354
	Reason	0.870	0.980	0.461	0.997	0.850	0.591	0.493	0.749	0.571	0.863	0.407	0.994	0.752	0.702	0.503
Gemini 3 Flash Preview	AO	0.644	0.649	0.427	0.879	0.814	0.616	0.512	0.649	0.424	0.702	0.431	0.947	0.729	0.702	0.441
	Reason	0.948	0.990	0.389	0.941	0.863	0.646	0.530	0.758	0.933	1.000	0.425	1.000	0.823	0.754	0.543
Gemini 3.1 Pro Preview	Reason	0.978	0.999	0.606	0.989	0.874	0.660	0.608	0.816	0.961	0.997	0.647	0.993	0.850	0.781	0.573
Human	AO	-	-	-	-	-	-	-	-	0.841	0.750	0.323	0.903	0.712	0.423	0.594

Table 2: Normalized scores per game for each model in Answer-Only (AO), Chain-of-Thought (CoT), or native Reasoning (Reason) regimes. **Wdl** = Wordle, **WC** = Word Chain, **SitP** = Situation Puzzles, **20Q** = 20 Questions, **WoF** = Wheel of Fortune, **20F** = 20 Features, and **Als** = Alias. The highest score in each column is shown in bold with a shaded background, while the second- and third-highest scores are shown in bold; tied scores share the same formatting. The 95% confidence intervals are reported in Appendix G.

4.3 Cross-Lingual Performance

The bilingual design lets us ask whether English and Russian measure the same capability, and where they diverge, on the same per-episode data underlying Table 2. For each model m and game g we form the English–Russian gap $\delta_{m,g} = s_{m,g}^{\text{EN}} - s_{m,g}^{\text{RU}}$, where s is the model’s mean score over that game’s episodes (a cell of Table 2). Per game, we then test whether these per-model gaps are centered on zero with a two-sided one-sample Wilcoxon signed-rank test over the models, reporting the mean gap with a bootstrap 95% CI and a sign test; Answer-Only and CoT/Reasoning configurations are tested separately (Appendix F).

Firstly, the two languages rank models almost identically (overall Spearman $\rho = 0.980$; per-game agreement in Table 7), so the Russian penalty shifts absolute score levels without reshuffling the ordering. Secondly, the largest and most consistent English advantages fall on the letter games Wordle and Wheel of Fortune (significant under both AO and Reasoning regimes; Word Chain only with reasoning). Situation Puzzles matches this in both

regimes; but overall the semantic games remain near parity between Russian and English scores (see Appendix Table 8). Thirdly, the Russian-language penalty is larger for weaker models and attenuated, but not eliminated, for strong models: on letter games, weak-tier Russian scores sit near the floor while strong models score high in both languages, whereas on semantic games the two languages are statistically indistinguishable for strong models (Table 4). Thus the cross-lingual penalty falls most heavily on letter-manipulation tasks and on weaker models.

Finally, a tokenization “token tax” is a plausible contributing factor. As a direct, if observational, probe of the Cyrillic-tokenization hypothesis, we measured output-side tokenizer fertility (completion_tokens per whitespace-separated word) on the reasoning trajectories generated during evaluation. (AO responses were excluded as too short for stable distributions, alongside Gemini and several GPT models whose reasoning traces are hidden via API). This is an observational measurement on model-generated text, not a controlled tokenizer ablation. Table 3 shows Russian is

Model	EN	RU	RU/EN
DeepSeek V4 Flash	1.77	2.56	1.45
DeepSeek V4 Pro	1.68	2.40	1.43
DeepSeek V3.1 Terminus	1.50	2.36	1.57
GPT 4.1	1.51	2.08	1.38
GPT OSS 120B	1.79	1.94	1.08
Qwen3 235B Instruct	1.67	2.72	1.63
Qwen3 235B Thinking	1.61	2.46	1.52
Qwen3 30B Instruct	1.56	2.53	1.62
Qwen3 30B Thinking	1.52	2.41	1.59
Qwen3.5 35B A3B	2.20	2.40	1.09

Table 3: Output-side tokenizer fertility (mean completion_tokens per whitespace-separated word) on reasoning trajectories. **EN/RU** are tokens per word; **RU/EN** is their ratio. Russian is consistently more token-expensive, markedly so for Qwen3 and DeepSeek. Answer-Only outputs and hidden-reasoning models are excluded (see text).

Games	Tier	EN	RU	EN-RU
Letter	Weak	0.365 [0.26,0.46]	0.227 [0.16,0.29]	0.138 [0.09,0.19]
	Strong	0.898 [0.87,0.92]	0.786 [0.73,0.86]	0.112 [0.06,0.15]
Semantic	Weak	0.371 [0.30,0.44]	0.328 [0.26,0.40]	0.043 [0.03,0.06]
	Strong	0.645 [0.63,0.67]	0.642 [0.62,0.68]	0.003 [-0.02,0.03]

Table 4: Absolute scores (mean with bootstrap 95% CI over configurations) by game type and model tier; tiers are the bottom/top third of model×mode configurations by overall capability. On letter games Russian trails English at every tier (gap CI excludes 0), while on semantic games the gap vanishes for strong models (gap CI includes 0).

consistently more token-expensive than English, with an RU/EN fertility ratio of 1.43–1.63 for the DeepSeek and Qwen3 models (with the newer Qwen3.5 35B A3B a low-fertility outlier at 1.09) but only 1.08–1.38 for the GPT models. Overall, we treat tokenization as a contributing factor rather than a proven cause, and leave controlled ablations to future work.

4.4 Judge Verification

We verified our LLM judge (DeepSeek V4 Flash) across Alias, 20 Features, and Situation Puzzles. We sampled 45 trajectories (AO and CoT) across eight models and compared turn-by-turn verdicts against three independent references: human annotations for Russian, GPT 5.4, and Gemini 3 Flash, with both reference models evaluated at temperature 0. Results are shown in Table 5.

For 20 Features and Alias, we measured *micro accuracy* (binary match rate). Agreement was near-perfect for 20 Features, while Alias had slightly lower scores due to the ambiguity of some questions about historical figures. For Situation Puzzles,

Game	Lang.	GPT 5.4	Gem. 3F	Ref.
Alias	RU	0.968	0.970	0.981
	EN	0.972	0.969	—
20 Feat.	RU	1.000	0.956	0.998
	EN	0.983	0.925	—
Sit. Puzz.	RU (str.)	0.69	0.75	0.72
	RU (soft)	0.75	0.80	0.90
	EN (str.)	0.80	0.82	—
	EN (soft)	0.87	0.90	—

Table 5: Agreement of DeepSeek V4 Flash judge against three reference sources across all three games. For Alias and 20 Features, exact match micro accuracy is reported. For Situation Puzzles, strict (str.) and soft (soft) micro accuracy are reported separately. Human reference annotations are available for Russian only.

we report *strict accuracy* (exact match across yes, no, and irrelevant) and *soft accuracy* (where disagreements between irrelevant and yes/no are not penalized). Because irrelevant signifies that a question does not actively confirm or deny a situation, soft accuracy measures whether a judge avoids directly misleading a player. Soft accuracy against human annotations reached 0.90, indicating that the judge usually provides directionally consistent feedback.

DeepSeek V4 Flash showed the highest agreement with the human reference. Its prompt was iteratively refined during development on a separate dataset that does not overlap with the benchmark evaluation data. Figure 8 shows a rare false-positive judgment.

4.5 Human Baseline

We benchmarked human performance on the same 100 Russian instances using a web dashboard mirroring the LLM evaluation setup (483 collected sessions, 476 retained for analysis, contributed by 15 paid native speakers, with 9–12 participants per game; Appendix I.5).

Results reveal a clear divide. Humans outperformed every AO configuration on Wordle (Human 0.841 vs. best AO 0.424) and Word Chain (0.750 vs. 0.702). On each of the other five games, the best AO result exceeded the human score, although this was not universal across AO configurations: on Situation Puzzles, 20 Features, and Alias, respectively, 5, 8, and 8 of the 12 AO configurations surpassed humans. Overall, the aggregated human score of 0.594 falls just below the best AO model (Gemini 3 Flash, 0.625) and below the top reasoning models.

5 Discussion

5.1 Wordle and Wheel of Fortune Complexity Levels

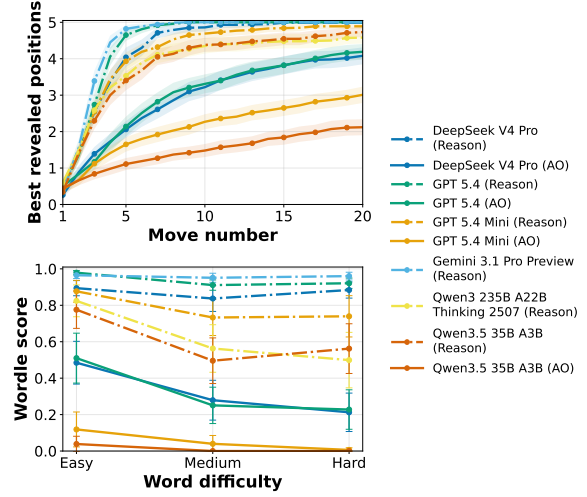
Wordle and Wheel of Fortune (WoF) admit a clear linguistic-rarity complexity axis: target-word frequency for Wordle and letter frequencies for WoF. We form three balanced tiers—Easy, Medium, and Hard—by bucketing words according to corpus frequency (Wordle) or an intrinsic score equal to the average marginal rarity of unique letters (WoF), with thresholds chosen to produce equally sized buckets (Appendix A.5.1). The right panels of Figures 1a and 1b show decreasing performance as complexity increases, illustrating the challenge of low-frequency vocabulary and letter distributions for current frontier LLMs.

5.2 Wordle Turns Saturation

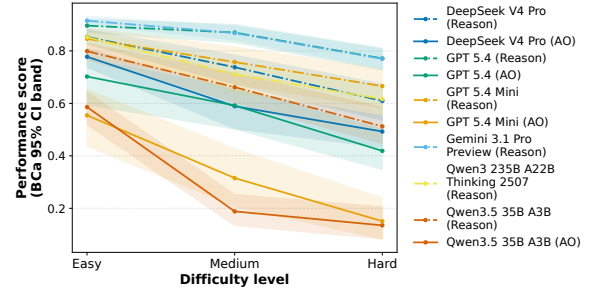
The left panel of Figure 1a shows an operational gap between tiers: weaker models can solve some Wordle instances but require more turns. Frontier models (Gemini 3.1 Pro Preview, GPT 5.4 Reason) consistently isolate the target word by turn 10, while lighter configurations (GPT 5.4 Mini AO, Qwen3.5 35B A3B AO) uncover only two or three correct positions before exhausting the turn limit. Trajectory slopes do not plateau at later turns, suggesting that weaker models could succeed under a larger step budget.

5.3 Wheel of Fortune Game Tactics

WoF efficiency depends on whether a player samples letters from a global corpus distribution (*marginal tactic*, derived from standard English/Russian noun lemma inventories) or from a distribution conditioned each turn on the remaining vocabulary consistent with the revealed pattern (*optimal conditional tactic*). Figure 2a shows human trajectories between weak AO models and frontier reasoning systems, mirroring the aggregated results in Table 2: top-tier models (e.g., Gemini 3.1 Pro Preview) track the conditional optimum, while Qwen3.5 35B A3B (AO) reverts to marginal sampling. Humans do not closely mirror the conditional optimum, suggesting an alternative heuristic driven by semantic word-level hypotheses rather than character-level frequency calculations. Cross-lingually, the marginal–conditional divergence is narrower in English (Figure 2b) than in Russian (Figure 2a), plausibly due to structural and dataset differences (alphabet size, lexical density) that af-



(a) Top: cumulative best number of correctly positioned letters observed by each move in Russian Wordle. Bottom: scores on Easy, Medium, and Hard target words. Shaded bands and error bars show 95% confidence intervals.



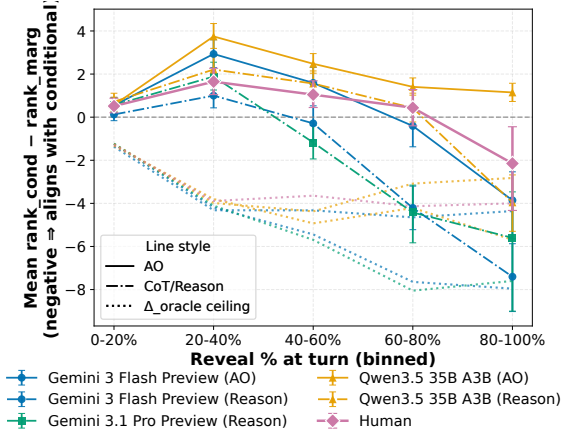
(b) Wheel of Fortune performance scores stratified by Easy, Medium, and Hard word complexity buckets for Russian language.

Figure 1: Analysis of performance scores and completion dynamics across different word complexity tiers in Wordle and Wheel of Fortune.

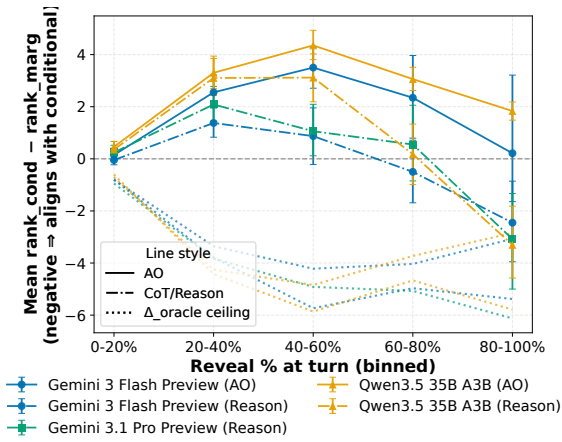
fect the baseline gap between marginal and conditional distributions; a fuller investigation is left for future work.

5.4 Cognitive Skills Correlation

To test whether multi-turn performance tracks distinct cognitive skills or generalized intelligence, we adapt the protocol of Momentè et al. (2025) and map correlations between game outcomes and 11 static cognitive batteries (Appendix H) across 12 frontier models from Qwen, DeepSeek, GPT, and Gemini. We evaluate in the Answer-Only (AO) regime to reduce prompt-format effects, with two API-driven adaptations: (i) substituting inaccessible log-likelihood rankings with direct multiple-choice string generation, and (ii) adding anti-reasoning constraints to suppress spontaneous CoT in highly aligned frontier models (Ap-



(a) LLM and human behavior in Russian Wheel of Fortune sessions.



(b) LLM behavior in English Wheel of Fortune sessions.

Figure 2: Difference between the rank of the model’s chosen letter in the optimal conditional distribution and its rank in the baseline marginal distribution across stages of word completion. Each line represents an LLM configuration; solid lines denote models operating in CoT/Reason modes, while dotted lines indicate Answer-Only (AO) modes. The reference line illustrates the theoretical baseline of consistently picking the top-ranked letter from the conditional distribution. Smaller differences indicate a more context-sensitive guessing tactic.

pendix H.2).

Figure 3 reports the resulting Kendall τ -b rank correlations and reveals a clear split. **Causal Reasoning** and **Pragmatics** achieve significance on all 7/7 games, followed by **Common-sense Reasoning** and **Planning** on 6/7 each. Because these single-turn tests target narrower abilities than the games, the correlations are consistent with shared model-level capabilities, but they do not establish that the same mechanisms are exercised across turns. Although Pragmatics is categorized

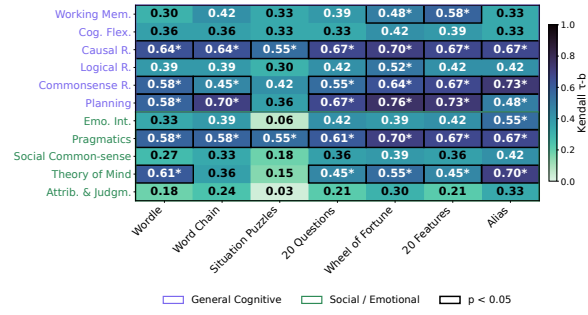


Figure 3: Kendall τ -b correlation matrix mapping 11 cognitive benchmark metrics against game performance scores (AO regime, English) across 12 frontier-class LLMs. Asterisks denote statistically significant correlations ($p < 0.05$).

under Social and Emotional Skills by [Momentè et al. \(2025\)](#), its uniform cross-game association in our results may reflect broader model capability rather than a domain-specific socio-emotional signal. Conversely, **Working Memory** achieves significance on only 2/7 games (Wheel of Fortune, 20 Features), and **Cognitive Flexibility** on none. In our suite, this pattern suggests that performance on the selected static working-memory and flexibility tests transfers less consistently to the games than causal reasoning, pragmatics, common-sense reasoning, or planning; it should not be interpreted as evidence that working memory or cognitive flexibility are generally unimportant for multi-turn reasoning.

Three metrics (Cognitive Flexibility, Social Common-sense, Attribution & Judgment) reach significance on no game, and on the game side **Situation Puzzles** is the clearest outlier: significant on only 2/11 batteries and lowest mean rank correlation in the suite, consistent with Situation Puzzles measuring capabilities not well covered by the selected static batteries. Overall, while absolute leaderboards agree at the extremes (Gemini 3 Flash and GPT 5.4 Nano anchor the top and bottom), our suite adds (i) wider score variance than saturation-prone static batteries, and (ii) per-game cognitive specialization: Wheel of Fortune and 20 Features emerge as the broadest proxies (7 and 6 significant skills), while Situation Puzzles is comparatively weakly associated with the battery suite, supporting the use of MTGBench as a complementary tool to static cognitive batteries—capturing interactive, multi-turn capabilities that saturation-prone single-turn tests leave underprobed—rather than a redundant proxy for them.

6 Related Work

Zhang et al. (2025) survey multi-turn LLM capabilities across memory, planning, instruction following, reasoning, and general conversation. Widely used benchmarks such as Multi-IF (He et al., 2024) and MathChat (Liang et al., 2024) concatenate independent single-turn tasks into a dialogue, so each turn remains evaluable in isolation. Our games instead create genuine multi-turn dependencies: the model must integrate prior feedback to make progress, a stricter test of adaptive reasoning.

A separate line of work evaluates LLMs through games. GAMA-Bench (Huang et al., 2024) studies multiplayer game-theoretic decision-making and ReTA’s Online Racing regime (Duan et al., 2024) evaluates head-to-head strategic play; such scores depend on other agents, and round-robin evaluation requires $O(n^2)$ matchups and is subject to positional bias. TextArena (Guertler et al., 2025) spans more than 57 text environments with model-versus-model and model-versus-human evaluation, and GameArena (Hu et al., 2025) uses three live computer games played with human participants. These efforts demonstrate the value of interactive gameplay but differ from our fixed puzzle suite, in which each model is evaluated independently on matched instances and open-form outcomes are handled by constrained rule-enforcer judges.

The closest procedural benchmark, MTR-Bench (Li et al., 2026), uses parametric generators and deterministic evaluators across 40 tasks, isolating induction, abduction, deduction, and planning through structured interactions; its authors note that this design does not assess reasoning expressed through natural dialogue. MTGBench is complementary: seven bilingual natural-language puzzle games, constrained judges where deterministic checking is insufficient, and a human baseline on matched instances.

TextWorld (Côté et al., 2019) provides turn-based text-game environments for training and evaluating RL agents, and LMRL Gym (Abdullahi et al., 2023) extends this to language-model policies across games and dialogue tasks, including Wordle and 20 Questions; both emphasize RL development rather than fixed, inference-time comparison of pretrained LLMs.

Other benchmarks test formal reasoning outside a communicative context: SATBench (Wei et al., 2025) (Boolean satisfiability), AQA-Bench (Yang et al., 2025) (search-algorithm ex-

ecution), WILT (Banatt et al., 2024) (hidden-rule discovery), LogicGame (Gui et al., 2025) (rule execution and planning), and PuzzlePlex (Long et al., 2025) (combinatorial, numerical, and spatial puzzles). These benchmarks target formal task states, whereas MTGBench centers on lexical and semantic interaction through words, letters, textual clues, and natural-language feedback.

Finally, recent work fine-tunes LLMs on game interactions: Tajwar et al. (2025) train a generally curious agent via RL on self-generated Wordle and 20 Questions data, and Playpen (Horst et al., 2025) applies SFT and GRPO to LLM-generated game dialogues. Both rely on LLM self-play, which risks distilling the generator’s biases (Li et al., 2025), highlighting the potential of game trajectories from algorithmic solvers for improving LLM quality.

7 Conclusions

We introduced **MTGBench**, a bilingual (English/Russian) suite of seven multi-turn puzzle games that probe distinct facets of LLM reasoning—deductive elimination, semantic chaining, lateral inference, logical consistency, and knowledge retrieval—under a unified evaluation protocol combining algorithmic checkers with constrained LLM-as-a-rule-enforcer judging.

Experiments across 16 frontier models reveal three robust patterns. First, model rankings are highly task-dependent: no single game predicts performance on the others, justifying multi-game evaluation. Second, the benefit of chain-of-thought is non-uniform—near-essential for letter-level manipulation (Wordle, Wheel of Fortune), but modest for knowledge-retrieval games. Third, cross-lingual disparities persist primarily in letter-level tasks and for smaller models, pointing to subword tokenization coverage as a key limitation.

Correlating game outcomes with standard cognitive batteries revealed heterogeneous, game-specific associations—most consistent for causal reasoning, pragmatics, planning, and common-sense reasoning—rather than a single underlying signal, positioning MTGBench as a complement to saturation-prone static cognitive tests rather than a proxy for them. We release code and data to facilitate future work on multilingual multi-turn evaluation at <https://github.com/skyberg11/MTGBench>.

8 Limitations

While MTGBench provides a reproducible framework for evaluating multi-turn reasoning in LLMs, several limitations should be considered when interpreting our findings.

Human Baseline Constraints. As detailed in Appendix I.6, the collection of human-playthrough data was constrained by several factors. The sample size is relatively small ($N = 56$ to 102 sessions per game) and consists entirely of native Russian speakers recruited from a single organization, limiting demographic and linguistic diversity.

Linguistic and Typological Scope. Although we present a bilingual benchmark (English and Russian), both languages belong to the Indo-European family and are well represented in the pre-training corpora of frontier models. The benchmark does not currently cover low-resource languages or languages with substantially different morphological structures (e.g., agglutinative or tonal languages). Consequently, the cross-lingual performance disparities observed in our study (such as the divergence in conditional tactics for Wheel of Fortune) may not fully generalize to a broader linguistic spectrum.

Dependence on LLM-as-a-Judge and Synthetic Data. To scale dataset generation for games like Situation Puzzles, 20 Features, and Alias, we utilized an ensemble of LLMs for data generation. Despite intersection filtering, embedding-based deduplication, and human validation, synthetic datasets may still contain structural biases or “distillation artifacts” inherited from the generator models. Additionally, while four of our games use purely algorithmic evaluators, the remaining three rely on DeepSeek V4 Flash acting as an LLM-as-a-rule-enforcer. As the error analysis in Section 4.4 shows, even a tightly constrained LLM judge can occasionally be passive or validate an unsupported answer, leading to false-positive wins.

API Constraints. Our evaluation of closed-weight models was conducted via commercial APIs (e.g., OpenRouter). This approach required methodological adaptations; for instance, we substituted traditional log-likelihood ranking with generation-based scoring for the cognitive benchmark suite (Appendix H.2).

Ecological Validity of Games. Finally, the games in our benchmark are closed-world, zero-sum environments with strict turn limits (typically 20-30 moves) and formalized feedback loops.

While showing game-specific associations with several cognitive skills (most consistently causal reasoning, pragmatics, planning, and common-sense reasoning), these highly structured tasks do not entirely capture the noise, ambiguity, and infinite action spaces inherent to real-world multi-step applications, such as autonomous web browsing or organic human-agent dialogue.

Acknowledgments

We thank Petr Ermakov and Tatiana Ilina for providing resources and organizational support. We are grateful to Eduard Klyshinsky and Alexander Panin for their careful reading of the manuscript and for their valuable comments and suggestions. Finally, we thank the Alice AI editorial team, whose participation in the game sessions made it possible to establish the human baseline for MTGBench.

References

- Marwa Abdulhai, Isadora White, Charlie Snell, Charles Sun, Joey Hong, Yuexiang Zhai, Kelvin Xu, and Sergey Levine. 2023. [Lmrl gym: Benchmarks for multi-turn reinforcement learning with language models](#). *Preprint*, arXiv:2311.18232.
- Kartikeya Badola, Jonathan Simon, Arian Hosseini, Sara Marie Mc Carthy, Tsendsuren Munkhdalai, Abhimanyu Goyal, Tomáš Kočiský, Shyam Upadhyay, Bahare Fatemi, and Mehran Kazemi. 2025. [Multi-turn puzzles: Evaluating interactive reasoning and strategic dialogue in llms](#). *Preprint*, arXiv:2508.10142.
- Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. [Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, page 7421–7454. Association for Computational Linguistics.
- Eryk Banatt, Jonathan Cheng, Skanda Vaidyanath, and Tiffany Hwu. 2024. [Wilt: A multi-turn, memorization-robust inductive logic benchmark for llms](#). *Preprint*, arXiv:2410.10998.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to evaluate chat-optimized language models as conversational agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.

- Aileen Cheng, Alon Jacovi, Amir Globerson, Ben Golan, Charles Kwong, Chris Alberti, Connie Tao, Eyal Ben-David, Gaurav Singh Tomar, Lukas Haas, Yonatan Bitton, Adam Bloniarz, Aijun Bai, Andrew Wang, Anfal Siddiqui, Arturo Bajuelos Castillo, Aviel Atias, Chang Liu, Corey Fry, and 46 others. 2025. [The facts leaderboard: A comprehensive benchmark for large language model factuality](#). *Preprint*, arXiv:2512.10791.
- Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. [Textworld: A learning environment for text-based games](#). *Preprint*, arXiv:1806.11532.
- Jinhao Duan, Shiqi Wang, James Diffenderfer, Lichao Sun, Tianlong Chen, Bhavya Kailkhura, and Kaidi Xu. 2024. [ReTA: Recursively thinking ahead to improve the strategic reasoning of large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2232–2246, Mexico City, Mexico. Association for Computational Linguistics.
- Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, Massachusetts.
- Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2021. [A framework for few-shot language model evaluation](#).
- Geoapify. 2026. World cities database: Cities, towns, villages and hamlets. <https://www.geoapify.com/download-all-the-cities-towns-villages/>. OpenStreetMap locality extracts distributed under ODbL; accessed 2026-08-18.
- Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, and 5 others. 2025. [Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai](#). *Preprint*, arXiv:2411.04872.
- Dongyu Gong, Xingchen Wan, and Dingmin Wang. 2024. [Working memory capacity of chatgpt: An empirical study](#). *Preprint*, arXiv:2305.03731.
- Yuling Gu, Oyvind Tafford, Hyunwoo Kim, Jared Moore, Ronan Le Bras, Peter Clark, and Yejin Choi. 2026. [Simpletom: Exposing the gap between explicit tom inference and implicit tom application in llms](#). *Preprint*, arXiv:2410.13648.
- Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. [Textarena](#). *Preprint*, arXiv:2504.11442.
- Jiayi Gui, Yiming Liu, Jiale Cheng, Xiaotao Gu, Xiao Liu, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. 2025. [LogicGame: Benchmarking rule-based reasoning abilities of large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1474–1491, Vienna, Austria. Association for Computational Linguistics.
- Yun He, Di Jin, Chaoqi Wang, Chloe Bi, Karishma Mandyam, Hejia Zhang, Chen Zhu, Ning Li, Tengyu Xu, Hongjiang Lv, Shruti Bhosale, Chenguang Zhu, Karthik Abinav Sankararaman, Eryk Helenowski, Melanie Kambadur, Aditya Tayade, Hao Ma, Han Fang, and Sinong Wang. 2024. [Multi-if: Benchmarking llms on multi-turn and multilingual instructions following](#). *Preprint*, arXiv:2410.15553.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). *Preprint*, arXiv:2103.03874.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. [Playpen: An environment for exploring learning through conversational interaction](#). *Preprint*, arXiv:2504.08590.
- Jennifer Hu, Sammy Floyd, Olessia Jouravlev, Evelina Fedorenko, and Edward Gibson. 2023. [A fine-grained comparison of pragmatic language understanding in humans and language models](#). *Preprint*, arXiv:2212.06801.
- Lanxiang Hu, Qiyu Li, Anze Xie, Nan Jiang, Ion Stoica, Haojian Jin, and Hao Zhang. 2025. [Gamearena: Evaluating llm reasoning through live computer games](#). In *The Thirteenth International Conference on Learning Representations*.
- Jen-tse Huang, Eric John Li, Man Ho Lam, Tian Liang, Wenxuan Wang, Youliang Yuan, Wenxiang Jiao, Xing Wang, Zhaopeng Tu, and Michael R. Lyu. 2024. [How far are we on the decision-making of LLMs? evaluating LLMs’ gaming ability in multi-agent environments](#). *Preprint*, arXiv:2403.11807.
- Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. [Livecodebench: Holistic and contamination free evaluation of large language models for code](#). *Preprint*, arXiv:2403.07974.
- Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik

- Narasimhan. 2024. [Swe-bench: Can language models resolve real-world github issues?](#) *Preprint*, arXiv:2310.06770.
- Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2024. [Cladder: Assessing causal reasoning in language models](#). *Preprint*, arXiv:2312.04350.
- Sean M Kennedy and Robert D Nowak. 2024. [Cognitive flexibility of large language models](#). In *ICML 2024 Workshop on LLMs and Cognition*.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. [Llms get lost in multi-turn conversation](#). *Preprint*, arXiv:2505.06120.
- Miaomiao Li, Hao Chen, Yang Wang, Tingyuan Zhu, Weijia Zhang, Kaijie Zhu, Kam-Fai Wong, and Jindong Wang. 2025. [Understanding and mitigating the bias inheritance in llm-based data augmentation on downstream tasks](#). *Preprint*, arXiv:2502.04419.
- Xiaoyuan Li, Keqin Bao, Yubo Ma, Moxin Li, Wenjie Wang, Rui Men, Yichang Zhang, Fuli Feng, and Dayiheng Liu. 2026. [Mtr-bench: A comprehensive benchmark for multi-turn reasoning evaluation](#). *Preprint*, arXiv:2505.17123.
- Zhenwen Liang, Dian Yu, Wenhao Yu, Wenlin Yao, Zhihan Zhang, Xiangliang Zhang, and Dong Yu. 2024. [Mathchat: Benchmarking mathematical reasoning and instruction following in multi-turn interactions](#). *Preprint*, arXiv:2405.19444.
- Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023. [Logiqa 2.0—an improved dataset for logical reasoning in natural language understanding](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2947–2962.
- Yitao Long, Yuru Jiang, Hongjun Liu, Yilun Zhao, Jingchen Sun, Yiqiu Shen, Chen Zhao, Arman Cohan, and Dennis Shasha. 2025. [Puzzleplex: Benchmarking foundation models on reasoning and planning with puzzles](#). *Preprint*, arXiv:2510.06475.
- N. V. Loukachevitch, G. Lashevich, A. A. Gerasimova, V. V. Ivanov, and B. V. Dobrov. 2016. Creating russian wordnet by conversion. In *Proceedings of International Conference on Computational Linguistics and Intellectual Technologies (Dialog 2016)*, pages 405–415.
- Filippo Momentè, Alessandro Suglia, Mario Giulianelli, Ambra Ferrari, Alexander Koller, Oliver Lemon, David Schlangen, Raquel Fernández, and Raffaella Bernardi. 2025. [Triangulating llm progress through benchmarks, games, and cognitive tests](#). *Preprint*, arXiv:2502.14359.
- Samuel J. Paech. 2024. [Eq-bench: An emotional intelligence benchmark for large language models](#). *Preprint*, arXiv:2312.06281.
- Long Phan, Alice Gatti, Nathaniel Li, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Dan Hendrycks, Ziwen Han, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, and 1081 others. 2026. [A benchmark of expert-level academic questions to assess ai capabilities](#). *Nature*, 649(8099):1139–1146.
- Jirui Qi, Shan Chen, Zidi Xiong, Raquel Fernández, Danielle Bitterman, and Arianna Bisazza. 2025. [When models reason in your language: Controlling thinking language comes at the cost of accuracy](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, page 20279–20296. Association for Computational Linguistics.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavathula, and Yejin Choi. 2019. [Winogrande: An adversarial winograd schema challenge at scale](#). *Preprint*, arXiv:1907.10641.
- Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. 2019. [Socialliqa: Commonsense reasoning about social interactions](#). *Preprint*, arXiv:1904.09728.
- David Schlangen. 2023. [Dialogue games for benchmarking language understanding: Motivation, taxonomy, strategy](#). *Preprint*, arXiv:2304.07007.
- Edward Loper Steven Bird, Ewan Klein. 2009. *Natural language processing with python*. O’Reilly Media, Inc.
- Fahim Tajwar, Yiding Jiang, Abitha Thankaraj, Sumaita Sadia Rahman, J Zico Kolter, Jeff Schneider, and Ruslan Salakhutdinov. 2025. [Training a generally curious agent](#). *Preprint*, arXiv:2502.17543.
- Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2025. [Judging the judges: Evaluating alignment and vulnerabilities in llms-as-judges](#). *Preprint*, arXiv:2406.12624.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [Mmlu-pro: A more robust and challenging multi-task language understanding benchmark](#). *Preprint*, arXiv:2406.01574.
- Koki Wataoka, Tsubasa Takahashi, and Ryokan Ri. 2025. [Self-preference bias in llm-as-a-judge](#). *Preprint*, arXiv:2410.21819.
- Anjiang Wei, Yuheng Wu, Yingjia Wan, Tarun Suresh, Huanmi Tan, Zhanke Zhou, Sanmi Koyejo, Ke Wang, and Alex Aiken. 2025. [Satbench: Benchmarking llms’ logical reasoning via automated puzzle generation from sat formulas](#). *Preprint*, arXiv:2505.14615.

Minghao Wu, Weixuan Wang, Sinuo Liu, Huifeng Yin, Xintong Wang, Yu Zhao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. [The bitter lesson learned from 2,000+ multilingual benchmarks](#). *Preprint*, arXiv:2504.15521.

Siwei Yang, Bingchen Zhao, and Cihang Xie. 2025. [Aqa-bench: An interactive benchmark for evaluating llms’ sequential reasoning ability](#). *Preprint*, arXiv:2402.09404.

Zheng-Xin Yong, M. Farid Adilazuarda, Jonibek Mansurov, Ruochen Zhang, Niklas Muennighoff, Carsten Eickhoff, Genta Indra Winata, Julia Kreutzer, Stephen H. Bach, and Alham Fikri Aji. 2025. [Crosslingual reasoning through test-time scaling](#). *Preprint*, arXiv:2505.05408.

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. 2025. [A survey on multi-turn interaction capabilities of large language models](#). *Preprint*, arXiv:2501.09959.

Huaixiu Steven Zheng, Swaroop Mishra, Hugh Zhang, Xinyun Chen, Minmin Chen, Azade Nova, Le Hou, Heng-Tze Cheng, Quoc V. Le, Ed H. Chi, and Denny Zhou. 2024. [Natural plan: Benchmarking llms on natural language planning](#). *Preprint*, arXiv:2406.04520.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhonghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. [Judging llm-as-a-judge with mt-bench and chatbot arena](#). *Preprint*, arXiv:2306.05685.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

A Task Formulations

In this section, we formally define the environments and rules for our games.

A.1 Word Chain

The game is played using a static snapshot of the Geoapify World Cities Database, an OpenStreetMap-derived locality database (Geoapify, 2026). The dataset is preprocessed by normalizing city names to ASCII and adding the available alternative names.

At each turn, a player must name a city that has not been used earlier in the current game. The city must also start with the effective last letter of the previously named city. During preprocessing, names are lowercased, non-letter characters are converted to separators, consecutive spaces are collapsed, and internal spaces are retained. For

Russian city names, the letter “ë” is replaced with “e”. If the normalized city name ends with “ь”, “Ъ”, or “ы”, the previous character is treated as the effective last letter.

A move is considered invalid if the proposed city is not present in the city database, has already been used, or does not start with the required letter. In any of these cases, the game terminates. The game is also stopped after a predefined maximum of 25 turns.

In the evaluation setting, the agent plays against a greedy baseline opponent. Before each move, the candidate list is deterministically reshuffled using a seed derived from the current set of used cities, and the opponent selects the first city in this order that satisfies all current constraints.

Dataset generation We collected city and town names, including their alternative names, and normalized them by converting characters outside the English alphabet (‘a–z’, ‘A–Z’) for English and the Russian alphabet (‘a–я’, ‘A–Я’) for Russian into separators, then collapsing consecutive spaces. The resulting list was used as the validation vocabulary for the game. We also sampled approximately 19K cities for English and 4K cities for Russian to serve as candidate guesses for the greedy baseline opponent, and selected 100 starting cities for each language. All city names produced by the LLMs were normalized in the same way before validation.

A.2 Wordle

The Wordle task is implemented as a finite-horizon word-guessing game with deterministic feedback. Each word is assigned a difficulty level—easy, medium, or hard—based on its frequency in large corpora. In the benchmark, the distribution of word difficulties is uniform across the three difficulty buckets. In our experiments, the vocabulary consists of five-letter nouns; nounhood is verified using WordNet (Steven Bird, 2009) for English and Ru-WordNet (Loukachevitch et al., 2016) for Russian.

On each turn, the agent proposes a five-letter word from the vocabulary. The environment then returns position-wise feedback for the proposed word. A letter is marked as correct if it matches the target letter in the same position. If the letter does not match in that position but occurs elsewhere in the target word, it is marked as present. Otherwise, it is marked as absent. For repeated letters, unmatched occurrences are checked independently against the full target string; target-letter

multiplicities are not exhausted.

The game continues until the agent guesses the target word or reaches the maximum number of allowed attempts. In our setting, the maximum number of attempts is 20. The episode is considered successful if the target word is guessed within this limit; otherwise, it is counted as a failure.

We stratified the sampling by difficulty level and selected 100 five-letter nouns for each language. The words were divided into three difficulty buckets.

A.3 Situation Puzzles

We treat the Lateral Thinking Puzzle task as an interactive guessing game with a hidden explanation. At the beginning of the game, the agent receives only a short and usually ambiguous scenario description. The full explanation is hidden from the agent, and the goal is to uncover enough information to explain what actually happened.

During the game, the agent can take two types of actions. First, it can ask yes/no-style questions to narrow down the possible interpretations of the scenario. The judge answers these questions using one of four labels: “Yes”, “No”, “Irrelevant”, or “Ambiguous question”. Second, the agent can submit a full explanation of the scenario as a solution attempt.

The judge has access to the hidden solution and evaluates the agent’s actions based on it. For questions, the judge provides the most appropriate answer according to the true explanation. For solution attempts, the judge returns a binary verdict indicating whether the proposed explanation is correct. A solution is accepted only if it covers the key fact required to explain the puzzle.

The goal of the agent is to solve the puzzle in as few steps as possible. The game ends either when the agent gives a correct explanation or when the maximum number of allowed steps is reached.

Dataset generation We generated the Situation Puzzles dataset synthetically using the two-stage pipeline detailed in Section B. The pipeline first constructs four semantically diverse attribute pools—Genre, Topic, Setting, and Twist Type. For each candidate puzzle, one sampled attribute is selected as a hard generation constraint; the other sampled attributes are not imposed on the model.

Each candidate is represented by four fields: the full story, the initial setup shown to the player, the hidden solution, and the key fact required for a cor-

rect answer. Candidates then pass format checks, embedding-based diversity selection, automated quality assurance, and final human validation. We retained 100 validated puzzles per language.

A.4 20 Questions

In the standard 20 Questions game, one player chooses a secret word and the other attempts to guess it by asking yes/no questions about the word’s properties. In typical LLM benchmarks, the judge selects a word and the tested model asks questions to guess it (Abdulhai et al., 2023; Tajwar et al., 2025). However, recent work (Badola et al., 2025) proposed an inverted variant in which the tested LLM chooses the secret word and the judge asks questions, thereby evaluating the model’s consistency. We adopt and extend this formulation: while in the original variant the model never explicitly names the secret word, in our setting the secret word is provided to the model in the system prompt, and the judge generates questions independently of the chosen word. This design isolates the model’s ability to remain consistent with a visible secret word under adversarial questioning.

The game episode is defined by a tuple (W, A, α, w_s, T) , where:

- $W = \{w_1, \dots, w_n\}$ is a pool of words, with $n \in [80, 100]$.
- $A = \{a_1, \dots, a_m\}$ is a set of attributes organized in a hierarchical taxonomy (root, mid-level, and leaf attributes). Attribute assignment respects transitive closure: if a word possesses a leaf attribute, it automatically inherits all ancestor attributes.
- $\alpha: W \rightarrow 2^A$ maps each word to its attribute set, with $|\alpha(w_i)| \in [5, 12]$ for each $w_i \in W$.
- $w_s \in W$ is the secret word, selected uniformly at random and revealed to the model in the system prompt.
- $T = 20$ is the maximum number of questions.

At the start of the game, both the word pool W and the complete attribute mapping α are provided to the model. The judge maintains a set of *remaining words* $R^{(0)} = W$ representing words consistent with all answers given so far.

At each step $t \in \{1, \dots, T\}$, the judge selects a question $q^{(t)}$ according to a rule-based strategy. Questions are of two types:

1. **Attribute question:** “Does your word have the attribute a_j ?” (in Russian, “У вашего слова есть атрибут a_j ?”) for some $a_j \in A$.
2. **Word guess:** “Is the word w_i ?” (in Russian, “Это слово w_i ?”) for some $w_i \in W$.

The question generation strategy is probabilistic:

- With probability 0.45, a *splitting* question is generated: the judge selects the attribute that splits the remaining words as close to evenly as possible:

$$a^* = \arg \min_{a \in A} \left| S_a - \frac{|R^{(t-1)}|}{2} \right|,$$

where $S_a = |\{w \in R^{(t-1)} : a \in \alpha(w)\}|$.

- With probability 0.20, an *exclusion* question is generated: the judge selects an attribute present only among words in $W \setminus R^{(t-1)}$, expecting a “no” answer.
- With probability 0.15, a *common attribute* question is generated: the judge selects an attribute shared by all words in $R^{(t-1)}$, expecting a “yes” answer.
- With probability 0.05, a *spontaneous word guess* is made.
- With probability 0.15, a *random attribute* question is made.
- If $|R^{(t-1)}| = 1$, the judge directly guesses the only remaining word.

The model responds with $r^{(t)} \in \{\text{yes}, \text{no}\}$. The judge then updates the remaining word set based on the question type and response.

For an **attribute question** about attribute a_j , let $W_{a_j} = \{w \in W : a_j \in \alpha(w)\}$. Then:

$$R^{(t)} = \begin{cases} R^{(t-1)} \cap W_{a_j}, & \text{if } r^{(t)} = \text{yes}, \\ R^{(t-1)} \setminus W_{a_j}, & \text{if } r^{(t)} = \text{no}. \end{cases} \quad (1)$$

For a **word guess** about word w_i :

- if $r^{(t)} = \text{yes}$ and $w_i \in R^{(t-1)}$, then $R^{(t)} = \{w_i\}$;
- if $r^{(t)} = \text{yes}$ and $w_i \notin R^{(t-1)}$, then $R^{(t)} = \emptyset$ (contradiction);
- if $r^{(t)} = \text{no}$, then $R^{(t)} = R^{(t-1)} \setminus \{w_i\}$.

A **contradiction** is detected when $w_s \notin R^{(t)}$, meaning that the model’s answers are no longer consistent with the secret word. This also occurs when the model confirms a word guess ($r^{(t)} = \text{yes}$) for $w_i \neq w_s$, or denies a guess ($r^{(t)} = \text{no}$) for $w_i = w_s$.

The game ends in one of the following outcomes:

- **Win:** The model answers “yes” (“да” in Russian) to a guess of w_s without any prior contradiction, or survives all T questions without contradiction.
- **Loss:** A contradiction is detected at any step t .

A.4.1 Synthetic dataset generation.

To isolate consistency from world knowledge, we use a fully synthetic dataset where both words and attributes are procedurally generated. Words are constructed via morphological composition: a root of 1-2 syllables (built from language-specific phoneme inventories for English and Russian) is optionally augmented with a prefix and/or suffix drawn from pools of English and Russian morphemes, producing pronounceable but meaningless strings. Attributes are organized in a three-level hierarchy (5 root categories, ~ 50 mid-level, ~ 100 leaf attributes), also generated as synthetic strings. Each word is assigned 5-12 attributes respecting transitive closure: if a word has a leaf attribute, it inherits all ancestors. Attribute sets are constrained to be unique across words.

Each episode samples 80–100 words and their attribute mappings from the generated pool, then selects one word uniformly at random as the secret word. The full dataset contains 400 episodes generated with a fixed random seed for reproducibility.

A.4.2 Performance Score.

The 20 Questions consistency score is calculated as

$$S_{20Q} = \begin{cases} 1, & \text{if the model wins,} \\ \frac{t-1}{T}, & \text{if the model loses,} \end{cases} \quad (2)$$

where t is the number of turns used and T is the maximum number of turns.

A.5 Wheel of Fortune

The Wheel of Fortune game is inspired by the TV show “Wheel of Fortune”¹. In this game, the agent

¹[https://en.wikipedia.org/wiki/Wheel_of_Fortune_\(American_game_show\)](https://en.wikipedia.org/wiki/Wheel_of_Fortune_(American_game_show))

must uncover a hidden word by guessing its letters one at a time or attempting to guess the entire word.

The game is defined by a tuple (Σ, w^*, E, T) , where:

- Σ is the alphabet of valid characters (English or Russian letters).
- $w^* = (w_1^*, \dots, w_N^*) \in \Sigma^N$ is the hidden target word of length N .
- $E = \{E_1, E_2, E_3\}$ is a set of equivalence classes over Σ that defines interchangeable letters for Russian only: $E_1 = \{\mathbb{I}, \mathbb{Й}\}$, $E_2 = \{\mathbb{E}, \mathbb{Ё}\}$, $E_3 = \{\mathbb{B}, \mathbb{Б}, \mathbb{Ь}\}$. Two letters ℓ_1, ℓ_2 are considered equivalent, denoted $\ell_1 \sim \ell_2$, if $\exists E_k \in E$ such that $\ell_1, \ell_2 \in E_k$.
- T is the maximum number of attempts allowed (default $T = 29$ for Russian and $T = 26$ for English, the number of distinct letter classes in Σ under $\sim$).

The agent observes a masked representation of the target word, where unrevealed positions are shown as underscores. Let $R^{(t)} \subseteq \Sigma$ denote the set of revealed letters after step t , with $R^{(0)} = \emptyset$. The displayed word at step t is:

$$d_i^{(t)} = \begin{cases} w_i^*, & \text{if } w_i^* \in R^{(t)} \text{ or} \\ & \exists \ell \in R^{(t)} : \ell \sim w_i^*, \\ _, & \text{otherwise,} \end{cases} \quad (3)$$

for $i = 1, \dots, N$.

At each step $t \in \{1, \dots, T\}$, the agent performs one of two actions:

1. **Letter guess** ($\ell^{(t)} \in \Sigma$): The agent proposes a single letter. Let $\bar{\ell}$ denote the canonical representative of $\ell^{(t)}$ under $\sim$. If $\bar{\ell}$ has already been guessed, the judge returns feedback that the letter was already guessed.

Otherwise, the letter is added to the set of guessed letters, and the outcome is determined as follows:

- If $\exists i$ such that $w_i^* \sim \ell^{(t)}$, the guess is *correct*: all matching positions are revealed, i.e., $R^{(t)} = R^{(t-1)} \cup \{w_i^* : w_i^* \sim \ell^{(t)}\}$.
- Otherwise, the guess is *incorrect*: $R^{(t)} = R^{(t-1)}$.

2. **Word guess** ($g^{(t)} \in \Sigma^*$): The agent proposes a complete word.

Every valid action (whether correct or incorrect) counts as one attempt. The game terminates under one of the following conditions:

- **Win by word guess**: $g^{(t)} = w^*$ for some step t .
- **Win by full reveal**: All positions are revealed, i.e., $d_i^{(t)} \neq _$ for all i .
- **Loss**: The maximum number of attempts T is reached without the word being guessed.

A.5.1 Dataset generation

We curated separate datasets for English and Russian. Target words are drawn from a curated vocabulary of Russian or English nouns and assigned a difficulty level based on letter rarity. The difficulty score of a word is defined as the mean rarity across its unique letters:

$$D(w) = \frac{1}{|\mathcal{U}(w)|} \sum_{\ell \in \mathcal{U}(w)} (100 - f(\ell)), \quad (4)$$

where $\mathcal{U}(w)$ is the set of unique letters in w and $f(\ell)$ is the corpus frequency (in percent) of letter ℓ . Words are partitioned into three difficulty levels (Easy, Medium, Hard), with score thresholds chosen to produce equally sized buckets.

A.5.2 Performance Score.

For Wheel of Fortune, we define a turn-dependent normalized reward $C(t)$ and assign this reward as the performance score only when the model wins:

$$C_{\text{WoF}}(t) = \text{clamp}\left(1 - \frac{t - 1 - T_{\text{floor}}}{T_{\text{max}} - T_{\text{floor}}}, 0, 1\right),$$

$$S_{\text{WoF}} = \begin{cases} C_{\text{WoF}}(t), & \text{if the model wins,} \\ 0, & \text{if the model loses.} \end{cases} \quad (5)$$

Here, $t - 1$ is the zero-based index of the terminal turn used by the scoring pipeline, $T_{\text{max}} = 29$ is the common normalization horizon for both languages, and $T_{\text{floor}} = |\mathcal{U}_{\sim}(w)|/2$ is half the number of unique letter-equivalence classes in the target word (with equality as the equivalence relation for English).

A.6 20 Features Game Formulation

In the 20 Features game, the tested model must identify a secret word based on a fixed set of 20 features revealed sequentially. Unlike the standard 20 Questions game where the model actively queries

properties, in this variant, the judge provides one feature and its value per turn, and the model must decide whether it has enough information to guess the word or needs more features. This evaluates the model’s ability to integrate sequential information, perform deductive reasoning, and accurately estimate its own knowledge state.

The game episode is defined by a tuple (W, F, ϕ, w_s, T) , where:

- $W = \{w_1, \dots, w_n\}$ is a pool of words.
- $F = \{f_1, \dots, f_{20}\}$ is a fixed set of 20 distinct attributes or features.
- $\phi: W \rightarrow V_1 \times \dots \times V_{20}$ maps each word to its feature values, where V_k is the set of possible values for feature f_k .
- $w_s \in W$ is the secret word, selected uniformly at random.
- $T = 20$ is the maximum number of moves.

At each step $t \in \{1, \dots, T\}$, the judge reveals the t -th feature and its value for the secret word: (f_t, v_t) , where $v_t = \phi(w_s)_t$. The model then takes an action $a^{(t)}$, which is either:

1. **Request next feature:** The model outputs a specific token (e.g., "nei") to proceed to the next step without guessing. This does not count as an incorrect answer.
2. **Word guess:** The model proposes a word $w^{(t)} \in W$.

If the model guesses a word, the judge checks if $w^{(t)} = w_s$.

- If $w^{(t)} = w_s$, the game ends in a **Win**.
- If $w^{(t)} \neq w_s$, the game continues.

If the model requests the next feature and $t = T$, the game ends in a **Loss** since no more features can be revealed and the turn limit is reached.

For 20 Features, we use $m_{20F} = 2$ as the estimated minimum number of steps required to gather enough information for a plausible guess. We define a turn-dependent normalized reward $C_{20F}(t)$ and assign it as the performance score only when the model wins:

$$C_{20F}(t) = \text{clamp}\left(1 - \frac{t - 1 - m_{20F}}{T - m_{20F}}, 0, 1\right),$$

$$S_{20F} = \begin{cases} C_{20F}(t), & \text{if the model wins,} \\ 0, & \text{if the model loses.} \end{cases} \quad (6)$$

Here, $t - 1$ is the zero-based index of the terminal turn used by the scoring pipeline, and $T = 20$ is the maximum number of turns.

A.7 Alias with Personalities Game Formulation

In the Alias with Personalities game, the tested model (the “Student”) must identify a secret public figure by interacting with a judge (the “Verifier”). The Verifier holds the full biography of the target person. At each turn, the Student can either ask a natural-language question about the person or attempt to guess their identity. This setup evaluates the model’s ability to formulate informative queries, deduce hidden information from biographical facts, and recognize when it has gathered sufficient evidence to make a confident guess.

The game episode is defined by a tuple (P, B, p_s, T) , where:

- $P = \{p_1, \dots, p_n\}$ is a pool of notable personalities.
- $B = \{b_1, \dots, b_n\}$ is a set of corresponding biographical texts, where b_i is the biography of p_i .
- $p_s \in P$ is the secret personality, selected uniformly at random. The Verifier is given b_s , while the Student only receives a generic instruction to identify the person.
- $T = 30$ is the maximum number of moves.

At each step $t \in \{1, \dots, T\}$, the Student takes an action $a^{(t)}$, which is either:

1. **Ask a question:** The Student generates a natural-language question $q^{(t)}$. The Verifier answers concisely (preferably yes/no) based strictly on the biography b_s . If the question cannot be answered from the biography, the Verifier declines to answer.
2. **Guess a personality:** The Student proposes a name $p^{(t)} \in P$.

If the Student makes a guess, the Verifier checks if $p^{(t)} = p_s$.

- If $p^{(t)} = p_s$, the game ends in a **Win**.
- If $p^{(t)} \neq p_s$, the game continues.

If the Student asks questions for all T turns without making a correct guess, the game ends in a **Loss**.

For Alias with Personalities, we use $m_{\text{Alias}} = 9$ as the estimated minimum number of steps required to gather enough information for a plausible guess. We define a turn-dependent normalized reward $C_{\text{Alias}}(t)$ and assign it as the performance score only when the model wins:

$$C_{\text{Alias}}(t) = \text{clamp}\left(1 - \frac{t - 1 - m_{\text{Alias}}}{T - m_{\text{Alias}}}, 0, 1\right),$$

$$S_{\text{Alias}} = \begin{cases} C_{\text{Alias}}(t), & \text{if the model wins,} \\ 0, & \text{if the model loses.} \end{cases} \quad (7)$$

Here, $t - 1$ is the zero-based index of the terminal turn used by the scoring pipeline, and $T = 30$ is the maximum number of turns.

B Situation Puzzles Dataset Generation Pipeline

To construct a diverse and high-quality dataset of lateral-thinking puzzles, we implemented a two-stage generation pipeline using multiple large language models (LLMs) and semantic filtering techniques. The process consists of (1) attribute generation and (2) puzzle synthesis, followed by automated and human quality assurance.

B.1 Attribute Generation

The first stage aims to create a comprehensive set of thematic attributes to guide the subsequent puzzle generation. We define four attribute categories: *Genres*, *Topics*, *Settings*, and *Twist Types*.

For each category C , we employ an ensemble of state-of-the-art models (Google Gemini 2.5 Pro Preview, DeepSeek Chat V3, and OpenAI GPT-5) to generate a large pool of candidate attributes $\mathcal{A}_{\text{raw}}$. To ensure semantic diversity and remove redundancy, we map each candidate $a \in \mathcal{A}_{\text{raw}}$ to a vector representation $v_a \in \mathbb{R}^d$ using a text embedding model. We then select a subset $\mathcal{A}_{\text{final}} \subset \mathcal{A}_{\text{raw}}$ of size K using the Farthest Point Sampling (FPS) algorithm.

Let S be the set of selected indices, initialized with a random index. At each step, we select the next item i^* that maximizes the minimum distance to the already selected set:

$$i^* = \arg \max_{i \notin S} \min_{j \in S} d(v_i, v_j) \quad (8)$$

where $d(x, y) = 1 - \frac{x \cdot y}{\|x\| \|y\|}$ is the cosine distance.

This process ensures that the selected attributes cover the semantic space as uniformly as possible.

B.2 Puzzle Synthesis

In the second stage, we generate the actual puzzles using the curated attributes. For each puzzle, we sample Genre, Topic, Setting, and Twist Type. To encourage creativity while maintaining coherence, we randomly select *one* attribute from the tuple to serve as a hard constraint in the generation prompt; the other three sampled attributes are not included in the prompt. Candidate puzzles in the released dataset were synthesized by Claude Opus 4.6, DeepSeek V4 Pro, Gemini 3.1 Pro Preview and GPT-5.4. The generation model is instructed to produce a structured output containing:

- **Full Story:** The complete narrative underlying the puzzle.
- **Setup:** The ambiguous scenario presented to the player.
- **Solution:** The complete explanation of the scenario.
- **Key Fact:** The essential fact that must be recovered for a correct solution.

B.2.1 Semantic Filtering and Quality Assurance

After generating a large batch of candidate puzzles, we apply a similar FPS-based filtering strategy. We compute embeddings for the concatenated string "Setup: [setup]. Solution: [solution]" for each puzzle. We then select a target number of puzzles that are maximally distant from each other in the semantic space, effectively removing duplicates and highly similar variations.

Each semantically selected puzzle then undergoes an automated quality check by a separate judge model (GPT-5 Mini). The judge evaluates the puzzle on a 1–10 scale based on logical consistency, solvability, and adherence to the “lateral thinking” format. Low scores and detected logical contradictions are used as filtering signals. Finally, a human annotator verifies that the setup is understandable, the solution is coherent, the key fact is well defined, and the puzzle can be solved through yes/no-style questioning. Only puzzles that pass this final validation are retained; the benchmark contains 100 puzzles per language.

C 20 Features Dataset Generation Pipeline

To ensure high-quality and unbiased word-feature mappings, we employ a multi-stage generation pipeline. First, we perform **Consensus-based lexical item selection** by prompting an ensemble of generative language models (YandexGPT 5 Lite, YandexGPT 5.1 Pro, AliceAI LLM, Gemini 2.5 Flash, DeepSeek v3.1 Terminus, GPT-4.1 Mini, GPT-4.1) to generate diverse, morphologically simple English words. We retain only the words that appear in the outputs of all models in the ensemble, yielding a high-reliability seed lexicon of approximately 1,200 words.

Next, we use **Multi-model attribute description generation** to assign values for the 20 fixed features. For each word and feature, we prompt multiple models to generate plausible values. We aggregate the candidate values across all models by taking their union to increase coverage and reduce single-model biases.

Finally, in the **Stochastic attribute value sampling** stage, the final feature vector for each word is constructed by independently sampling one value per feature from the aggregated candidate set using a uniform distribution. All random operations use a fixed seed for full reproducibility.

D Alias with Personalities Dataset Generation Pipeline

The dataset is constructed using a two-stage pipeline to ensure diversity and factual accuracy. In the **Celebrities names generation** stage, we prompt an ensemble of generative language models (YandexGPT 5 Lite, YandexGPT 5.1 Pro, AliceAI LLM, Gemini 2.5 Flash, DeepSeek v3.1 Terminus, GPT-4.1 Mini, GPT-4.1) to generate lists of widely recognizable individuals across science, arts, politics, and sports. To mitigate model-specific biases and hallucinations, we apply intersection filtering, retaining only personalities that appear in all generated lists.

In the **Wikipedia biography retrieval** stage, we extract biographical data for each selected personality using the `wikipediaapi` Python package. We extract the introductory summary, key infobox fields (occupation, dates, nationality, notable works), and section headers. Entries with insufficient content (e.g., < 300 characters or missing structured data) are filtered out. We also exclude living individuals with sensitive non-public infor-

mation to comply with privacy guidelines. The final dataset consists of high-quality personality-biography pairs in structured JSON format.

E Model Evaluation Details

We evaluated a total of 16 models spanning four major model families: Qwen, DeepSeek, GPT, and Gemini. The full list of evaluated models is provided in Table 6.

Inference formats. Each model was evaluated under two inference formats where applicable. In the *Answer Only* (AO) format, the model was instructed to produce a direct answer without any intermediate reasoning steps. For models with native reasoning capabilities, reasoning was explicitly disabled during inference. Models for which reasoning could not be disabled were not evaluated in the AO format. In the *Chain-of-Thought / Reasoning* (CoT/Reasoning) format, the evaluation procedure differed depending on the model type. For standard instruction-tuned models without built-in reasoning, a chain-of-thought prompt explicitly asked the model to reason before providing a final answer. For models with native reasoning capabilities, reasoning mode was enabled during inference. For GPT and Gemini models, we set `reasoning_effort` to "none" for AO and "medium" for CoT/Reasoning. For the other model families, reasoning was enabled or disabled through the corresponding API flag rather than `reasoning_effort`.

API and inference settings. All models were queried through the OpenRouter API without pinning a specific upstream provider. For OpenAI and Gemini models, requests were issued with `service_tier='flex'` to reduce inference costs. Temperature, `top_p`, maximum output tokens, and all other generation parameters were left at their API defaults. Prior to the main evaluation, we manually inspected samples from each model to verify that outputs were coherent and well formed and that the requested reasoning mode was applied as expected.

F Cross-Lingual Statistical Tests

This appendix reports the supporting cross-lingual tests summarized in Section 4.3: rank agreement between the English and Russian leaderboards (Table 7), and per-game across-model test of the EN–RU level gap (Table 8). For every game

Family	Model	AO	CoT / Reasoning
Qwen	Qwen3 30B A3B Instruct 2507	✓	✓ (CoT)
	Qwen3 30B A3B Thinking 2507		✓ (Reasoning)
	Qwen3 235B A22B Instruct 2507	✓	✓ (CoT)
	Qwen3 235B A22B Thinking 2507		✓ (Reasoning)
	Qwen3.5 35B A3B	✓	✓ (Reasoning)
DeepSeek	DeepSeek V3.1 Terminus	✓	✓ (Reasoning)
	DeepSeek V4 Flash	✓	✓ (Reasoning)
	DeepSeek V4 Pro	✓	✓ (Reasoning)
GPT	GPT OSS 120B		✓ (Reasoning)
	GPT 4.1	✓	✓ (CoT)
	GPT 5.4 Nano	✓	✓ (Reasoning)
	GPT 5.4 Mini	✓	✓ (Reasoning)
	GPT 5.4	✓	✓ (Reasoning)
Gemini	Gemini 3.1 Flash Lite	✓	✓ (Reasoning)
	Gemini 3 Flash Preview	✓	✓ (Reasoning)
	Gemini 3.1 Pro Preview		✓ (Reasoning)

Table 6: List of evaluated models and their supported inference formats.

Game	Group	Spearman ρ	Kendall τ
Word Chain	letter	0.945	0.831
Wordle	letter	0.958	0.850
Wheel of Fortune	letter	0.961	0.831
20 Features	semantic	0.942	0.820
20 Questions	semantic	0.836	0.683
Alias	semantic	0.892	0.739
Situation Puzzles	semantic	0.638	0.455
Total	Total	0.980	0.910

Table 7: Rank agreement between the EN and RU model leaderboards (over model $\times$ mode configurations), per game and overall (all $p < 0.001$). High values indicate the two languages rank models the same way despite the absolute-level gap on letter games.

we collapse each model to a single EN–RU gap (mean_EN – mean_RU) and run a one-sample Wilcoxon signed-rank and sign test over the independent models, doing so separately for Answer-Only and Reasoning/CoT inference formats; per-game Wilcoxon p are Holm-corrected across the game $\times$ regime family.

G Winrates and Confidence Intervals

Table 10 presents the win rates of the evaluated models across our games. In this setting, only the binary outcome of each game is considered—whether the model succeeded within the allotted number of turns—without accounting for the num-

ber of turns required to achieve a win. Confidence intervals for the scores reported in Table 2 are provided in Table 11 and estimated via bootstrapping over episodes completed by the analysis cutoff. For the Total score, episodes are resampled separately within each game using the corresponding observed sample size, after which the seven game-level scores are averaged. Table 9 provides the per-cell breakdown of episodes excluded at the analysis cutoff.

H Cognitive Skills Benchmark

H.1 Benchmark Suite

We reproduce all 11 cognitive benchmarks from [Momentè et al. \(2025\)](#). Table 12 provides an overview.

H.2 Differences from the Original Protocol

We introduce two deliberate deviations from [Momentè et al. \(2025\)](#). Each is forced by our implementation constraints (API-only access to frontier models) rather than by a desire to deviate from the original methodology, and each is disclosed below so that the per-cell correlation numbers in Figure 3 can be interpreted in context.

(D1) Generation-based scoring instead of log-likelihood ranking. The original work used the LM Evaluation Harness ([Gao et al., 2021](#)) for sev-

Game	Favor EN	Mean Δ [95% CI]	Wilc. p	Holm p	Sign p
<i>Answer-Only (12 configs)</i>					
Wordle	12/12	+0.200 [0.135, 0.307]	< 0.001	0.005	< 0.001
Wheel of Fortune	12/12	+0.155 [0.115, 0.184]	< 0.001	0.005	< 0.001
Word Chain	9/12	+0.078 [0.028, 0.134]	0.027	0.134	0.146
Situation Puzzles	10/12	+0.095 [0.059, 0.134]	0.002	0.015	0.039
Alias	12/12	+0.060 [0.038, 0.093]	< 0.001	0.005	< 0.001
20 Questions	9/12	+0.052 [0.017, 0.100]	0.034	0.134	0.146
20 Features	2/12	-0.045 [-0.077, -0.014]	0.027	0.134	0.039
Letter (all)	12/12	+0.144 [0.102, 0.185]	< 0.001	—	< 0.001
Semantic (all)	11/12	+0.041 [0.022, 0.060]	0.003	—	0.006
Overall	12/12	+0.085 [0.062, 0.105]	< 0.001	—	< 0.001
<i>CoT/Reasoning (16 configs)</i>					
Wordle	16/16	+0.214 [0.146, 0.297]	< 0.001	< 0.001	< 0.001
Wheel of Fortune	16/16	+0.097 [0.077, 0.128]	< 0.001	< 0.001	< 0.001
Word Chain	13/16	+0.117 [0.066, 0.179]	0.002	0.015	0.021
Situation Puzzles	14/16	+0.101 [0.061, 0.143]	< 0.001	0.005	0.004
Alias	8/16	+0.009 [-0.010, 0.034]	0.669	1.000	1.000
20 Questions	10/16	+0.006 [-0.030, 0.038]	0.597	1.000	0.454
20 Features	1/16	-0.073 [-0.092, -0.050]	< 0.001	0.001	< 0.001
Letter (all)	16/16	+0.143 [0.104, 0.186]	< 0.001	—	< 0.001
Semantic (all)	10/16	+0.011 [-0.011, 0.030]	0.404	—	0.454
Overall	13/16	+0.067 [0.041, 0.094]	< 0.001	—	0.021

Table 8: EN vs. RU level, tested the same way for every game and stratified by inference regime: Answer-Only and CoT/Reasoning configs form separate blocks. Within a block each model is collapsed to a single EN—RU gap for the game, then a one-sample Wilcoxon signed-rank and sign test are run over the independent models. Per-game Wilcoxon p are Holm-corrected across the game $\times$ regime family. $\Delta = \text{EN} - \text{RU}$; a positive value favors English.

eral benchmarks (LogiQA 2.0, WinoGrande, Social IQA, EQ-Bench), evaluating them via **log-likelihood scoring**—the model scores the likelihood of each answer option, and the highest-likelihood option is selected. Log-likelihood scoring is not available for frontier models served behind chat APIs. We therefore reformulate all single-turn multiple-choice benchmarks as **generation-based tasks**: the model receives a prompt with the question and labelled answer options (e.g., A/B/C/D) and is instructed to respond with the letter of its chosen answer. We then parse the response with per-benchmark regular expressions to recover the choice. This makes our absolute scores on multiple-choice benchmarks not directly comparable with the paper’s numbers—our scoring is more permissive on responses where a model emits a verbose answer that still surfaces a single letter.

(D2) Hardened anti-reasoning prompts on selected benchmarks. In an initial round of generation runs we observed that several frontier models produce a chain-of-thought trace by default before emitting their final answer, even though we did not request CoT. This was concentrated in three benchmarks: planning (NATURAL PLAN—every model verbose, two Qwen models with responses up to 81,460 and 104,839 characters), causal_reasoning (CLADDER—

Qwen-235B emitted reasoning on 2.3% of items, max 20,432 chars), and logical_reasoning (LogiQA 2.0—DeepSeek-V3.1 on 0.6% of items, max 9,966 chars).

We therefore added explicit no-reasoning instructions to the three affected benchmarks. The hardened system prompts include sentences such as “*You answer with ONLY the single word ‘yes’ or ‘no’ – no reasoning, no explanation, no draft attempts, no commentary, no other text.*”. Each benchmark additionally appends an IMPORTANT: tail suffix after the user question restating the same constraint. After this change, response verbosity on yes/no and multiple-choice benchmarks dropped to a median of 1–3 characters across all 12 models in our evaluation.

H.3 Benchmarks Score

We present the resulting scores for all cognitive benchmarks in Table 13 and visualize them following Momentè et al. (2025) in Figure 4.

I Human Evaluation Protocol

This appendix gives the full setup behind the Human row in Table 2 and Section 4.5. The goal of the human evaluation is to anchor model scores against a non-LLM baseline on *exactly the instances LLMs were scored on*, so that any LLM-vs-human comparison reduces to “what did a human do on the same target word / puzzle that the model just played?”.

I.1 Dashboard Architecture

Human play-throughs were collected through a purpose-built web application (FastAPI + React + SQLite). The dashboard reuses the benchmark’s game implementations verbatim—so the same process_player_move, win/loss bookkeeping, and (where applicable) LLM-as-judge calls are used for both humans and LLMs. Each game session writes a result JSON in the same format as the benchmark.

Deterministic game pool. At startup the backend replays the benchmark’s per-game seeded sampling with random seed=42 and number of games=100, yielding the same instance pool per game type as a benchmark run with the same seed. The 100 indices per game are distributed across players to maximize unique-target coverage.

Game (total excluded)	Model	Mode	Lang.	Samples excluded	Samples used
Wordle (5)	Qwen3.5 35B A3B	Reason	EN	2	98
	Qwen3.5 35B A3B	Reason	RU	3	97
Word Chain (9)	DeepSeek V3.1 Terminus	Reason	EN	7	93
	DeepSeek V4 Pro	Reason	EN	1	99
	Gemini 3.1 Pro Preview	Reason	RU	1	99
Situation Puzzles (2)	DeepSeek V3.1 Terminus	Reason	EN	1	99
	Qwen3 235B A22B Thinking 2507	Reason	EN	1	99
20 Questions (0)		No incomplete episodes			
Wheel of Fortune (3)	DeepSeek V4 Pro	Reason	EN	1	99
	GPT 5.4 Mini	Reason	EN	1	99
	Qwen3.5 35B A3B	Reason	RU	1	99
20 Features (2)	DeepSeek V4 Flash	Reason	RU	1	99
	GPT OSS 120B	Reason	RU	1	99
Alias (22)	Gemini 3.1 Flash Lite	Reason	RU	2	98
	Qwen3 235B A22B Thinking 2507	Reason	EN	14	86
	Qwen3 30B A3B Instruct 2507	CoT	RU	3	97
	Qwen3 30B A3B Thinking 2507	Reason	EN	2	98
	Qwen3.5 35B A3B	Reason	RU	1	99
All games (392 model–game–language cells)				43	–

Table 9: Breakdown of the 43 model episodes excluded from the reported analyses because they were incomplete at the analysis cutoff. Counts in parentheses are game-level totals. “Samples used” is the number of completed episodes used for the affected model–game–language samples.

Per-game max attempts. These match the benchmark exactly: Wordle, Situation Puzzles, 20 Questions, 20 Features: 20 turns; Word Chain: 25; Alias: 30; Wheel of Fortune: 29.

LLM-as-judge. Games whose oracle uses an LLM (Situation Puzzles, Alias, 20 Features) are configured with the same judge model the benchmark used at temperature 0.

I.2 Prompt Rendering: What Humans See

The system prompt is the same as the prompt used in the LLM benchmark, except for small edits that accommodate differences between LLM and human players. We replace LLM-specific output-format instructions for UI affordances.

I.3 Participants

Players were paid native Russian speakers recruited from the authors’ affiliated organisation. Participation was voluntary (participants self-selected from an internal channel announcement). No demographic data (age, education, NLP-familiarity) was collected. There was no tutorial or practice round; players read the in-game system prompt (the same one the LLM reads) and started playing immediately. We did not enforce a no-external-tools policy.

We did not cap the number of games per player. Coverage across the seven games is uneven: the heaviest contributors logged 8–18 games on the games they self-selected into; many players played only one game type.

I.4 Aggregation: Per-Target Averages

Each LLM configuration was scheduled for exactly one episode per pool index (one play of the target); the reported aggregation uses only episodes completed by the analysis cutoff. Humans, in contrast, sometimes played the same target multiple times because of the dashboard’s distribution algorithm—e.g., in 20 Questions, 8 out of 52 unique targets were played by two or more different humans. To make the Human row comparable cell-for-cell with the LLM rows we aggregate multiple human sessions on the same target into a single Human episode:

- $\text{won}_T^{\text{Human}} = \mathbf{1}[\overline{\text{won}_T} \geq 0.5]$ (majority vote; ties resolved as a win).
- $\text{turns}_T^{\text{Human}} = \text{round}(\overline{\text{turns}_T})$.

The per-game performance score for the Human row is then computed by the same per-episode formula as for the LLM rows (Appendix A).

Model	Mode	English (EN)								Russian (RU)							
		Wdl	WC	SitP	20Q	WoF	20F	Als	Total	Wdl	WC	SitP	20Q	WoF	20F	Als	Total
Qwen3 30B A3B Instruct 2507	AO	0.040	0.000	0.470	0.160	0.620	0.410	0.280	0.283	0.010	0.000	0.340	0.150	0.650	0.290	0.250	0.241
	CoT	0.850	0.040	0.670	0.260	0.960	0.430	0.310	0.503	0.460	0.000	0.390	0.310	0.770	0.390	0.340	0.380
Qwen3 30B A3B Thinking 2507	Reason	0.960	0.700	0.590	0.990	0.850	0.480	0.500	0.724	0.600	0.300	0.500	0.960	0.800	0.560	0.520	0.606
Qwen3 235B A22B Instruct 2507	AO	0.370	0.000	0.790	0.230	0.960	0.540	0.480	0.481	0.090	0.000	0.680	0.250	0.860	0.530	0.370	0.397
	CoT	0.960	0.160	0.790	0.520	1.000	0.590	0.320	0.620	0.720	0.020	0.640	0.370	0.990	0.700	0.450	0.556
Qwen3 235B A22B Thinking 2507	Reason	0.980	0.950	0.778	0.970	1.000	0.830	0.581	0.870	0.770	0.660	0.680	0.960	1.000	0.900	0.560	0.790
Qwen3.5 35B A3B	AO	0.200	0.000	0.610	0.140	0.990	0.560	0.440	0.420	0.040	0.000	0.420	0.070	0.750	0.500	0.330	0.301
	Reason	0.980	0.710	0.800	0.990	1.000	0.910	0.730	0.874	0.835	0.540	0.690	0.950	1.000	0.920	0.758	0.813
DeepSeek V3.1 Terminus	AO	0.840	0.010	0.820	0.580	1.000	0.630	0.630	0.644	0.090	0.040	0.750	0.570	0.960	0.660	0.420	0.499
	Reason	0.980	0.892	0.768	0.990	1.000	0.860	0.870	0.909	0.970	0.390	0.800	0.920	1.000	0.880	0.690	0.807
DeepSeek V4 Flash	AO	0.460	0.140	0.830	0.450	1.000	0.650	0.660	0.599	0.250	0.150	0.700	0.290	0.930	0.700	0.390	0.487
	Reason	0.960	0.200	0.800	0.990	1.000	0.810	0.730	0.784	0.910	0.180	0.750	0.820	1.000	0.889	0.700	0.750
DeepSeek V4 Pro	AO	0.810	0.370	0.830	0.840	1.000	0.730	0.650	0.747	0.560	0.110	0.700	0.540	0.990	0.850	0.540	0.613
	Reason	1.000	0.545	0.840	1.000	0.850	0.770	0.858	0.990	0.270	0.790	0.890	1.000	0.950	0.740	0.804	0.804
GPT OSS 120B	Reason	0.970	0.720	0.670	0.850	1.000	0.810	0.680	0.814	0.750	0.640	0.680	0.920	0.990	0.848	0.620	0.778
GPT 4.1	AO	0.550	0.060	0.910	0.480	0.990	0.780	0.670	0.634	0.250	0.050	0.730	0.460	0.960	0.920	0.690	0.580
	CoT	0.900	0.780	0.920	0.600	1.000	0.770	0.760	0.819	0.300	0.200	0.750	0.420	1.000	0.850	0.710	0.604
GPT 5.4 Nano	AO	0.090	0.000	0.590	0.060	0.610	0.380	0.240	0.281	0.000	0.000	0.420	0.020	0.480	0.380	0.180	0.211
	Reason	0.970	0.750	0.710	0.450	0.980	0.690	0.480	0.719	0.860	0.580	0.710	0.560	1.000	0.690	0.550	0.707
GPT 5.4 Mini	AO	0.250	0.030	0.740	0.250	0.880	0.640	0.380	0.453	0.130	0.020	0.660	0.130	0.570	0.680	0.330	0.360
	Reason	1.000	0.390	0.770	0.780	1.000	0.840	0.490	0.753	0.930	0.160	0.780	0.930	0.990	0.940	0.600	0.761
GPT 5.4	AO	0.840	0.190	0.880	0.950	1.000	0.850	0.520	0.749	0.570	0.240	0.780	0.870	0.980	0.820	0.580	0.691
	Reason	0.990	0.890	0.890	0.980	1.000	0.860	0.680	0.897	1.000	0.940	0.870	0.960	1.000	0.950	0.820	0.934
Gemini 3.1 Flash Lite	AO	0.520	0.160	0.730	0.920	0.990	0.720	0.680	0.674	0.200	0.070	0.760	0.870	0.880	0.750	0.570	0.586
	Reason	0.980	0.960	0.770	0.990	1.000	0.850	0.750	0.900	0.750	0.700	0.760	0.990	1.000	0.940	0.755	0.842
Gemini 3 Flash Preview	AO	0.860	0.410	0.780	0.830	0.990	0.870	0.730	0.781	0.650	0.450	0.770	0.930	0.990	0.930	0.610	0.761
	Reason	1.000	0.970	0.780	0.900	1.000	0.830	0.790	0.896	1.000	1.000	0.840	1.000	1.000	0.940	0.810	0.941
Gemini 3.1 Pro Preview	Reason	1.000	0.990	0.930	0.980	1.000	0.880	0.860	0.949	1.000	0.980	0.920	0.990	1.000	0.990	0.880	0.966
Human	AO	-	-	-	-	-	-	-	-	0.988	0.466	0.500	0.885	1.000	0.778	0.434	0.721

Table 10: Win rates per game for each model in Answer-Only (AO), Chain-of-Thought (CoT), or native Reasoning (Reason) regimes. **Wdl** = Wordle, **WC** = Word Chain, **SitP** = Situation Puzzles, **20Q** = 20 Questions, **WoF** = Wheel of Fortune, **20F** = 20 Features, and **Als** = Alias. The highest score in each column is shown in bold with a shaded background, while the second- and third-highest scores are shown in bold; tied scores share the same formatting.

I.5 Exclusion Criteria

Of the 483 collected sessions, 476 were retained for analysis. During quality control, we excluded seven early sessions from one participant—four Word Chain sessions immediately followed by three 20 Questions sessions—as test runs rather than study observations. The Word Chain sessions ended after 3–6 moves, and each 20 Questions session lasted under five seconds and ended after 2–4 turns. This exclusion was applied before per-target aggregation; no other collected sessions were excluded.

I.6 Limitations of the Human Baseline

- **Small, non-representative sample.** Participants were recruited from a single organization; the sample is not balanced for age, NLP-familiarity, or general gaming experience.
- **No CoT scaffold.** Humans played without any chain-of-thought elicitation; the Human row is best interpreted as a “no-CoT human” baseline. We do not run a parallel “CoT human” condition.

- **Uneven coverage.** Player participation was voluntary and not balanced across games; small-sample uncertainty on the Human row is therefore wider than on the LLM rows ($N = 52$ –100 aggregated targets per game).

J Failure-Mode Analysis

Aggregate win rates summarize *how often* a model fails, but not *how*. For each of our games we therefore classify every trajectory into a small set of game-specific failure modes, then report the percentage of analyzed games in which each mode was triggered. Detection is rule-based and uses the same ground-truth signals the game judges already record, so the classification is reproducible from the released result JSONs alone. For the Human rows, these trajectory-level percentages are computed over the retained sessions for the corresponding game (476 across all seven games), before the per-target aggregation used for the main score.

J.1 20 Questions

Losses in 20 Questions follow from incorrect answers to word or attribute questions, as defined

Model	Mode	English (EN)										Russian (RU)									
		Wdl	WC	SitP	20Q	WoF	20F	Als	Total	Wdl	WC	SitP	20Q	WoF	20F	Als	Total	Wdl	WC	SitP	20Q
Qwen3 30B A3B Instruct 2507	AO	[0.004, 0.040]	[0.066, 0.098]	[0.084, 0.173]	[0.280, 0.414]	[0.240, 0.350]	[0.180, 0.323]	[0.093, 0.203]	[0.158, 0.197]	[0.000, 0.060]	[0.076, 0.115]	[0.029, 0.089]	[0.280, 0.415]	[0.221, 0.330]	[0.138, 0.276]	[0.082, 0.199]	[0.139, 0.175]				
	CoT	[0.624, 0.760]	[0.189, 0.265]	[0.448, 0.622]	[0.416, 0.555]	[0.646, 0.730]	[0.184, 0.324]	[0.151, 0.292]	[0.417, 0.467]	[0.226, 0.368]	[0.215, 0.284]	[0.194, 0.336]	[0.466, 0.605]	[0.390, 0.514]	[0.172, 0.314]	[0.162, 0.300]	[0.298, 0.346]				
Qwen3 30B A3B Thinking 2507	Reason	[0.806, 0.894]	[0.798, 0.907]	[0.402, 0.574]	[0.979, 1.000]	[0.581, 0.701]	[0.258, 0.417]	[0.239, 0.383]	[0.617, 0.665]	[0.373, 0.534]	[0.578, 0.705]	[0.310, 0.475]	[0.942, 0.993]	[0.482, 0.607]	[0.297, 0.453]	[0.236, 0.378]	[0.502, 0.553]				
	AO	[0.152, 0.279]	[0.175, 0.237]	[0.466, 0.602]	[0.360, 0.504]	[0.504, 0.590]	[0.282, 0.443]	[0.225, 0.366]	[0.345, 0.393]	[0.016, 0.072]	[0.098, 0.158]	[0.303, 0.428]	[0.371, 0.516]	[0.358, 0.462]	[0.286, 0.446]	[0.170, 0.310]	[0.259, 0.304]				
Qwen3 235B A22B Instruct 2507	CoT	[0.763, 0.858]	[0.423, 0.554]	[0.531, 0.689]	[0.562, 0.721]	[0.719, 0.772]	[0.301, 0.456]	[0.191, 0.350]	[0.539, 0.590]	[0.437, 0.588]	[0.284, 0.371]	[0.375, 0.528]	[0.491, 0.635]	[0.626, 0.698]	[0.414, 0.578]	[0.226, 0.371]	[0.447, 0.498]				
	Reason	[0.866, 0.938]	[0.929, 0.992]	[0.541, 0.690]	[0.946, 0.996]	[0.803, 0.839]	[0.507, 0.653]	[0.323, 0.494]	[0.736, 0.776]	[0.557, 0.712]	[0.752, 0.870]	[0.404, 0.551]	[0.935, 0.994]	[0.688, 0.756]	[0.592, 0.728]	[0.277, 0.420]	[0.640, 0.686]				
Qwen3 5.5B A3B	AO	[0.066, 0.167]	[0.003, 0.010]	[0.211, 0.324]	[0.279, 0.406]	[0.482, 0.559]	[0.271, 0.425]	[0.174, 0.305]	[0.239, 0.280]	[0.003, 0.042]	[0.036, 0.062]	[0.114, 0.214]	[0.218, 0.326]	[0.252, 0.355]	[0.241, 0.387]	[0.134, 0.260]	[0.164, 0.203]				
	Reason	[0.852, 0.929]	[0.792, 0.897]	[0.489, 0.621]	[0.955, 1.000]	[0.773, 0.815]	[0.548, 0.678]	[0.362, 0.495]	[0.715, 0.753]	[0.545, 0.691]	[0.693, 0.824]	[0.301, 0.428]	[0.943, 0.993]	[0.617, 0.693]	[0.627, 0.752]	[0.374, 0.507]	[0.623, 0.667]				
DeepSeek V3.1 Terminus	AO	[0.564, 0.702]	[0.307, 0.384]	[0.285, 0.394]	[0.608, 0.765]	[0.642, 0.701]	[0.361, 0.524]	[0.355, 0.515]	[0.483, 0.532]	[0.024, 0.098]	[0.243, 0.345]	[0.285, 0.401]	[0.590, 0.752]	[0.418, 0.507]	[0.386, 0.546]	[0.259, 0.422]	[0.351, 0.399]				
	Reason	[0.866, 0.938]	[0.907, 0.979]	[0.342, 0.446]	[0.963, 1.000]	[0.801, 0.844]	[0.526, 0.669]	[0.474, 0.603]	[0.728, 0.764]	[0.776, 0.865]	[0.596, 0.732]	[0.278, 0.390]	[0.997, 0.975]	[0.675, 0.740]	[0.539, 0.676]	[0.254, 0.403]	[0.625, 0.664]				
DeepSeek V4 Flash	AO	[0.241, 0.394]	[0.388, 0.512]	[0.406, 0.532]	[0.519, 0.675]	[0.615, 0.676]	[0.343, 0.501]	[0.341, 0.480]	[0.447, 0.497]	[0.110, 0.238]	[0.360, 0.492]	[0.265, 0.384]	[0.417, 0.565]	[0.409, 0.514]	[0.410, 0.569]	[0.174, 0.307]	[0.345, 0.395]				
	Reason	[0.837, 0.926]	[0.472, 0.596]	[0.457, 0.598]	[0.978, 1.000]	[0.811, 0.858]	[0.476, 0.621]	[0.353, 0.480]	[0.658, 0.699]	[0.698, 0.812]	[0.475, 0.596]	[0.331, 0.451]	[0.825, 0.929]	[0.670, 0.736]	[0.539, 0.678]	[0.328, 0.459]	[0.589, 0.634]				
DeepSeek V4 Pro	AO	[0.510, 0.652]	[0.612, 0.741]	[0.396, 0.533]	[0.820, 0.930]	[0.693, 0.742]	[0.404, 0.561]	[0.349, 0.495]	[0.581, 0.629]	[0.275, 0.412]	[0.354, 0.492]	[0.181, 0.289]	[0.579, 0.735]	[0.576, 0.658]	[0.560, 0.703]	[0.259, 0.400]	[0.437, 0.487]				
	Reason	[0.921, 0.960]	[0.771, 0.873]	[0.511, 0.647]	[1.000, 1.000]	[0.827, 0.873]	[0.522, 0.663]	[0.438, 0.571]	[0.739, 0.777]	[0.838, 0.903]	[0.512, 0.647]	[0.399, 0.533]	[0.868, 0.961]	[0.693, 0.767]	[0.641, 0.767]	[0.387, 0.516]	[0.656, 0.698]				
GPT OSS 120B	Reason	[0.855, 0.932]	[0.818, 0.915]	[0.438, 0.611]	[0.796, 0.921]	[0.713, 0.779]	[0.496, 0.644]	[0.330, 0.469]	[0.674, 0.721]	[0.538, 0.691]	[0.768, 0.882]	[0.323, 0.465]	[0.936, 0.987]	[0.613, 0.699]	[0.516, 0.656]	[0.316, 0.467]	[0.613, 0.659]				
	CoT	[0.646, 0.770]	[0.914, 0.968]	[0.618, 0.733]	[0.655, 0.797]	[0.729, 0.787]	[0.462, 0.621]	[0.424, 0.562]	[0.673, 0.717]	[0.117, 0.232]	[0.511, 0.648]	[0.418, 0.552]	[0.546, 0.684]	[0.613, 0.685]	[0.531, 0.675]	[0.368, 0.501]	[0.481, 0.529]				
GPT 5.4 Nano	AO	[0.024, 0.099]	[0.032, 0.060]	[0.162, 0.263]	[0.220, 0.319]	[0.283, 0.405]	[0.160, 0.297]	[0.072, 0.168]	[0.160, 0.197]	[0.000, 0.000]	[0.023, 0.046]	[0.110, 0.201]	[0.173, 0.247]	[0.164, 0.274]	[0.228, 0.397]	[0.050, 0.137]	[0.125, 0.160]				
	Reason	[0.799, 0.893]	[0.873, 0.949]	[0.373, 0.515]	[0.592, 0.727]	[0.733, 0.810]	[0.389, 0.549]	[0.175, 0.293]	[0.599, 0.644]	[0.595, 0.725]	[0.801, 0.890]	[0.347, 0.472]	[0.664, 0.794]	[0.683, 0.754]	[0.419, 0.583]	[0.220, 0.354]	[0.571, 0.618]				
GPT 5.4 Mini	AO	[0.095, 0.215]	[0.412, 0.513]	[0.331, 0.466]	[0.393, 0.536]	[0.519, 0.624]	[0.368, 0.526]	[0.179, 0.316]	[0.366, 0.415]	[0.030, 0.110]	[0.243, 0.337]	[0.246, 0.356]	[0.314, 0.434]	[0.275, 0.405]	[0.391, 0.548]	[0.142, 0.276]	[0.267, 0.312]				
	Reason	[0.886, 0.935]	[0.595, 0.726]	[0.372, 0.505]	[0.753, 0.887]	[0.797, 0.855]	[0.508, 0.654]	[0.270, 0.426]	[0.635, 0.680]	[0.731, 0.838]	[0.437, 0.565]	[0.357, 0.484]	[0.884, 0.974]	[0.711, 0.793]	[0.632, 0.761]	[0.296, 0.435]	[0.617, 0.661]				
GPT 5.4	AO	[0.541, 0.678]	[0.583, 0.690]	[0.425, 0.533]	[0.940, 0.992]	[0.729, 0.778]	[0.518, 0.660]	[0.249, 0.395]	[0.602, 0.645]	[0.274, 0.419]	[0.420, 0.565]	[0.322, 0.442]	[0.844, 0.948]	[0.517, 0.619]	[0.525, 0.672]	[0.251, 0.376]	[0.491, 0.539]				
	Reason	[0.921, 0.975]	[0.924, 0.981]	[0.436, 0.551]	[0.936, 1.000]	[0.864, 0.900]	[0.567, 0.712]	[0.357, 0.490]	[0.747, 0.782]	[0.924, 0.953]	[0.924, 0.986]	[0.379, 0.496]	[0.926, 0.993]	[0.821, 0.866]	[0.682, 0.804]	[0.419, 0.550]	[0.753, 0.787]				
Gemini 3.1 Flash Lite	AO	[0.282, 0.434]	[0.520, 0.636]	[0.284, 0.409]	[0.936, 0.985]	[0.708, 0.770]	[0.397, 0.549]	[0.373, 0.527]	[0.536, 0.581]	[0.064, 0.165]	[0.344, 0.462]	[0.252, 0.362]	[0.848, 0.948]	[0.493, 0.602]	[0.444, 0.599]	[0.282, 0.429]	[0.426, 0.472]				
	Reason	[0.822, 0.905]	[0.943, 0.994]	[0.392, 0.527]	[0.982, 1.000]	[0.827, 0.871]	[0.520, 0.659]	[0.417, 0.565]	[0.730, 0.767]	[0.490, 0.648]	[0.805, 0.908]	[0.341, 0.473]	[0.964, 1.000]	[0.716, 0.784]	[0.637, 0.762]	[0.431, 0.573]	[0.663, 0.706]				
Gemini 3.1 Flash Preview	AO	[0.575, 0.711]	[0.578, 0.718]	[0.359, 0.494]	[0.820, 0.927]	[0.780, 0.837]	[0.542, 0.683]	[0.434, 0.584]	[0.625, 0.673]	[0.350, 0.498]	[0.631, 0.765]	[0.362, 0.501]	[0.895, 0.977]	[0.694, 0.758]	[0.634, 0.762]	[0.360, 0.523]	[0.601, 0.649]				
	Reason	[0.918, 0.965]	[0.964, 0.999]	[0.321, 0.459]	[0.896, 0.972]	[0.844, 0.878]	[0.571, 0.715]	[0.462, 0.592]	[0.740, 0.776]	[0.902, 0.951]	[1.000, 1.000]	[0.357, 0.493]	[1.000, 1.000]	[0.803, 0.841]	[0.690, 0.808]	[0.476, 0.603]	[0.766, 0.799]				
Gemini 3.1 Pro Preview	Reason	[0.965, 0.985]	[0.994, 1.000]	[0.540, 0.667]	[0.961, 1.000]	[0.856, 0.889]	[0.588, 0.728]	[0.540, 0.669]	[0.799, 0.833]	[0.946, 0.971]	[0.987, 1.000]	[0.576, 0.709]	[0.958, 1.000]	[0.829, 0.869]	[0.722, 0.829]	[0.507, 0.631]	[0.814, 0.844]				
	Human	AO								[0.793, 0.876]	[0.673, 0.819]	[0.224, 0.431]	[0.805, 0.963]	[0.679, 0.743]	[0.333, 0.521]	[0.136, 0.298]	[0.566, 0.623]				

Table 11: Confidence intervals for scores from Table 2 for each model in Answer-Only (AO), Chain-of-Thought (CoT), or native Reasoning (Reason) regimes. **Wdl** = Wordle, **WC** = Word Chain, **SitP** = Situation Puzzles, **20Q** = 20 Questions, **WoF** = Wheel of Fortune, **20F** = 20 Features, and **Als** = Alias.

in Section A.4. We use the following four failure modes:

- **FalsePosAttr**: model answered “yes” to an attribute the secret word does not have – the model invented a property.
- **FalseNegAttr**: model answered “no” to an attribute the secret word does have – the model denied a true property.
- **WrongYes**: model said “yes” to a final word guess that was not the secret word.
- **WrongNo**: model said “no” to a final word guess that *was* the secret word.

Table 14 gives the breakdown for Russian and English.

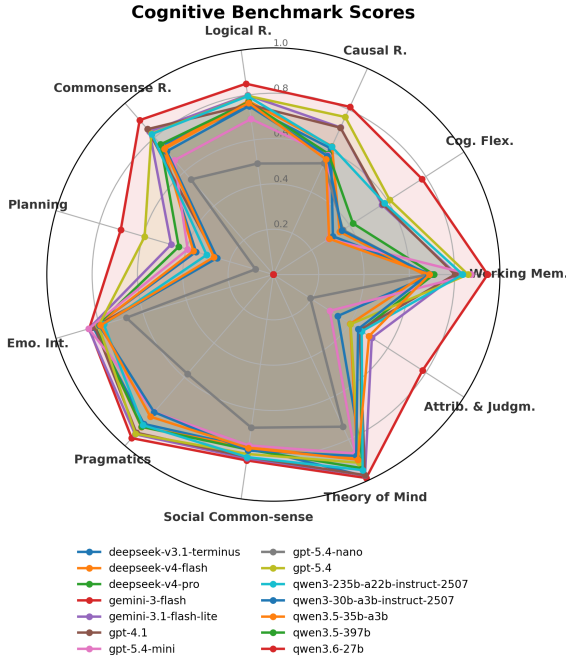
J.2 Wheel of Fortune

Unlike 20 Questions, Wheel of Fortune does not record a single failure reason: a game can end in a loss for many reasons and a model may also waste turns without losing. We walk each trajectory and tag every move against six independent detectors. A game can be flagged in multiple categories, so column sums are not partitions of the loss count; instead they characterize the *distribution of mistake types* a model makes during play.

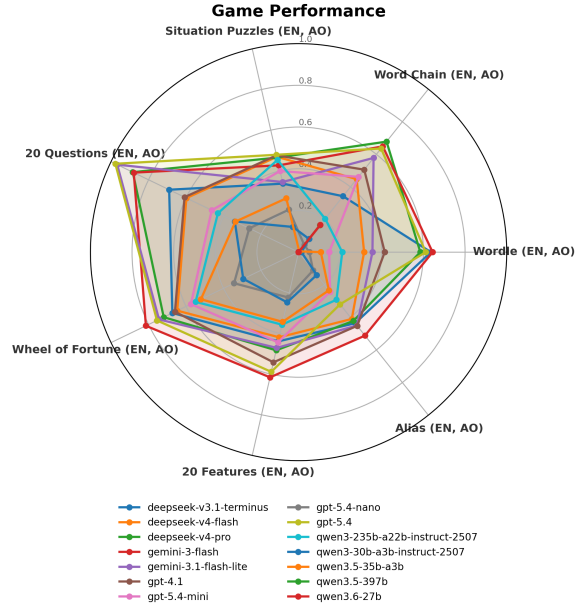
- **WrongLen**: word guess whose length differs from the target. The pattern (e.g. _ _ B A K A) makes target length explicit, so this is a basic constraint violation.

- **PatViol.**: length-correct word guess that contradicts a letter the pattern had already revealed (e.g. guessing *ПОБЕДА* against C _ B A K A). Reconstructed by replaying prior letter guesses against the target.
- **NonWord**: length- and pattern-consistent guess that is not a real word according to the dictionary. We use RuWordNet (Loukachevitch et al., 2016) for Russian and NLTK WordNet (Steven Bird, 2009) for English; words absent from these resources are flagged.
- **RepLetter**: a single-letter guess that has already been tried in the same game. Wastes the turn (the game rejects it without penalty) but is a strong signal that the model is not tracking its own move history.
- **FmtFail**: response that the game parser could not recognize (missing OTBET: marker, mixed alphabets, empty answer, ...).
- **NoWordGuess**: lost game in which the model never once attempted a word-level guess – it spent its entire budget on single letters.

Table 15 gives the Russian and English breakdowns. Three patterns recur across the leaderboard: (i) smaller or weaker models accumulate **RepLetter** flags at rates of 30% and above, indicating insufficient move-history tracking; (ii) **NonWord** guesses persist at 5–50% even for strong reasoning models – length and pattern constraints are easier than lexical validity; (iii) the human baseline



(a) Cognitive skill scores per model.



(b) Game scores per model (AO, English).

Figure 4: Cognitive skills and game performance radar plots.

shows near-zero constraint violations (**WrongLen**, **PatViol**.) but a small **RepLetter** tail consistent with players occasionally losing track of which letters they had already tried.

J.3 Wordle

Wordle guesses are checked before color feedback is produced. We tag three behaviors that indicate failures to respect the game state or its lexical constraints:

- **WordRepeat**: the model submits a word that it has already guessed in the same episode. Repeating a guess cannot reveal new information and indicates incomplete use of the dialogue history.
- **InvalidWord**: the submitted string has the required length and alphabet but is rejected by the language-specific noun lexicon (NLTK WordNet for English and RuWordNet for Russian). This category therefore reflects both model errors and the coverage limits of the lexical resources.
- **WrongLen**: the submitted guess does not have the same number of letters as the target, even though the required length is fixed by the game.

Table 16 reports the percentage of English and Russian sessions in which each behavior occurs at least once. The categories are not mutually exclusive: a trajectory may contain, for example, both a wrong-length attempt and a later repeated word. Answer-only configurations exhibit frequent lexical rejections and repetitions, particularly in Russian. Reasoning generally reduces repetitions and wrong-length guesses, but dictionary rejection remains visible for several strong configurations. The non-zero **InvalidWord** rate for humans also cautions against interpreting this detector as a pure hallucination measure.

J.4 Situation Puzzles

Situation Puzzles permits both information-seeking questions and free-form solution attempts. We report one trajectory-level behavior: **NotQuiteRight**, triggered when the player proposes a solution that the judge rejects because it differs from the hidden explanation or omits a key fact. The detector marks attempted but incomplete explanations; it is not itself evidence of a judge error, and it can occur in a trajectory that is later solved successfully.

Table 17 reports the share of trajectories containing at least one such near-miss. The rates vary substantially across models and inference modes, without a uniform English–Russian ordering. They should not be read as an inverse leaderboard: a

model that attempts solutions earlier or more often has more opportunities to trigger the detector, whereas a passive model may fail without ever submitting a near-miss.

J.5 20 Features

Table 18 documents instances of persistent error, in which the model reiterated identical incorrect responses across the game session despite receiving explicit corrective feedback. The rates vary considerably across models and inference modes, and can be read as an indirect indicator of how well the model follows the instructions.

J.6 Alias

Table 19 documents instances of multiple final hypotheses, illustrating a dichotomy in the models’ strategic behavior. Some models adopt a conservative strategy that delays commitment until the target is isolated to a small, high-confidence subset, whereas others adopt an aggressive strategy that commits to early guesses absent sufficient entity-specific inquiry.

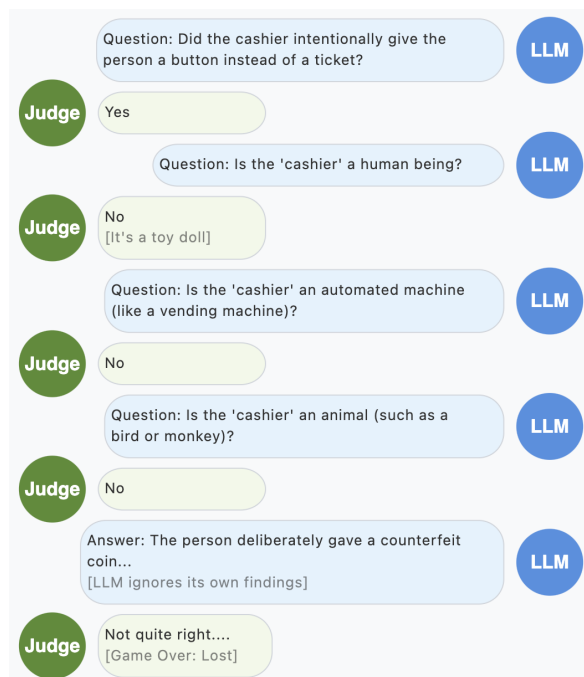


Figure 5: An example of a game lost by Gemini 3.1 Pro Preview. In this puzzle, the model is asked why a person who hands a coin to a ticket window receives a button in return yet remains satisfied (the true solution is that this is a child’s game with a toy train station). Although the model deduces that the “cashier” is neither human, machine, nor animal, its final guess does not use these established facts.

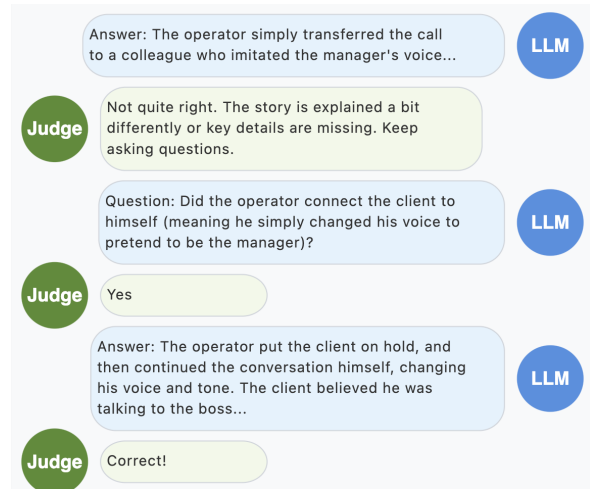


Figure 6: An example of a successful iterative correction by the LLM player. In this puzzle, an operator connects a client to a manager when no one else is in the building (the true solution is that the operator put the call on hold and changed their own voice). The model initially guesses that a colleague assisted. After the judge rejects this near-miss, the model adjusts its reasoning, asks a pivotal question, and gives the correct solution on the following turn.

J.7 Word Chain

Word Chain ends as soon as the player submits an invalid city. We separate these terminal errors into three categories:

- **RepCity**: the submitted city has already appeared in the episode, either as a player or environment move.
- **WrongLetter**: the city does not begin with the final valid letter required by the preceding city.
- **NotFound**: the normalized name is absent from the static validation snapshot derived from the Geoapify World Cities Database.

Table 20 reports the percentage of English and Russian trajectories ending in each category. Because the first invalid city terminates the episode, these terminal categories are mutually exclusive; trajectories that reach the turn limit or end for another reason contribute to none of them. Wrong initial letters dominate many answer-only configurations, while several models instead fail mainly by reusing a city. Reasoning reduces these errors for most, but not all, configurations. **NotFound** is usually less common for models than the other two categories; its comparatively high human rate

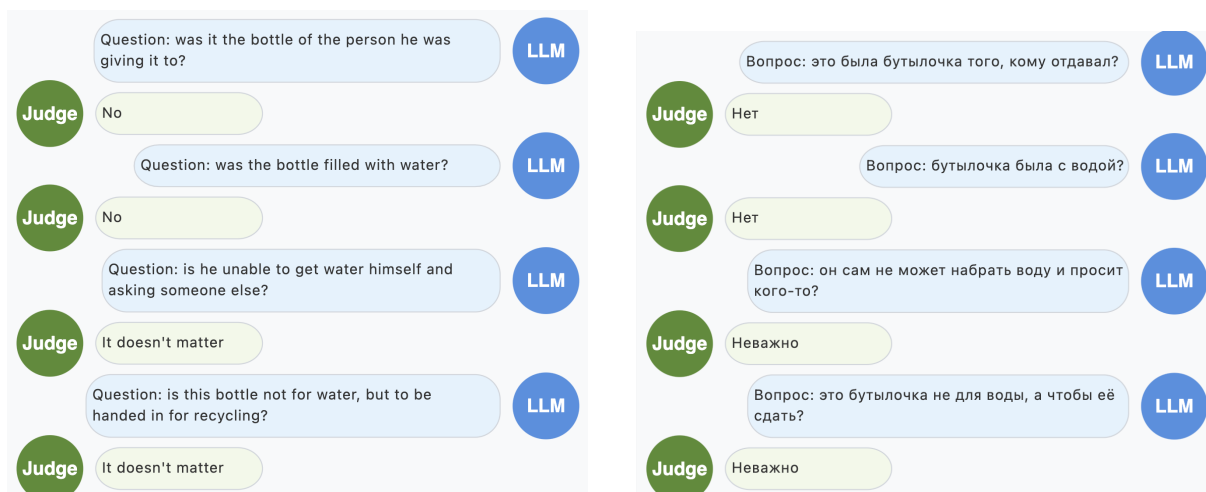


Figure 7: An example of a game derailed by a human reading error and judge passivity. In this puzzle, a boy gives his “bun” (*булочка* in Russian) to a classmate who helps him open the packaging. The human player misreads “bun” as “bottle” (*бутылочка*). The LLM judge responds with “No” or “It doesn’t matter” without rejecting the false premise, allowing the player to continue investigating an object that is not present in the scenario. **(Left)** The English translation. **(Right)** The original dialogue in Russian.

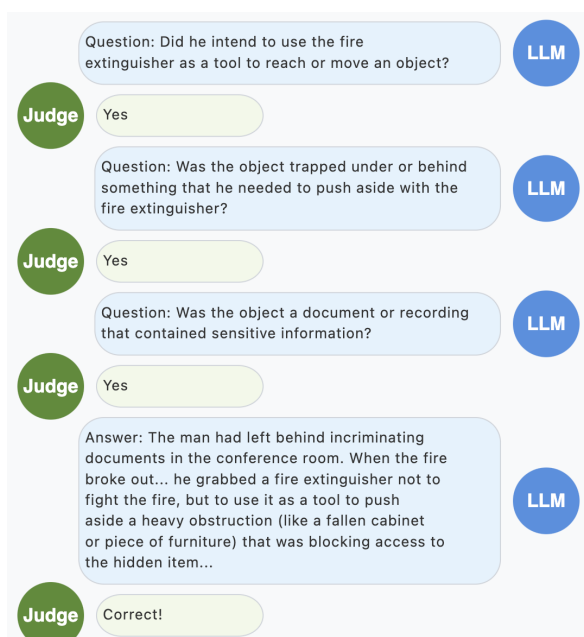


Figure 8: An example of a false-positive win caused by an LLM judge error. In this puzzle, a man grabs a fire extinguisher during a fire evacuation (the true solution is that he started the fire to steal documents and used the extinguisher as a disguise). The judge validates the player’s unsupported hypothesis about pushing a heavy obstruction and ultimately accepts an incorrect solution.

in Russian also makes it sensitive to database coverage, alternative names, and normalization rather than reasoning alone.

K Data and Ethics Statements

K.1 Data Statement

This work introduces a bilingual multi-turn game benchmark in English and Russian. The benchmark uses both automatically collected lexical resources and newly generated task datasets.

For the Wordle and Wheel of Fortune tasks, the target and validation vocabularies consist of nouns in the corresponding language. English lexical validation is based on NLTK WordNet, and Russian lexical validation is based on RuWordNet. Words were selected automatically according to task-specific constraints such as part of speech, word length, and frequency- or rarity-based difficulty stratification. Since these lexical resources aim to represent natural language usage, they may contain words referring to ethnic, national, religious, gender, or other social groups, as well as words related to sensitive, offensive, taboo, or otherwise potentially harmful topics. The presence of such items reflects the contents of the underlying lexical resources and does not represent the views, endorsement, or intent of the authors.

For the Word Chain task, we use a static database of real-world city and town names obtained from Geoapify, including available alternative names. The names are automatically normalized and used

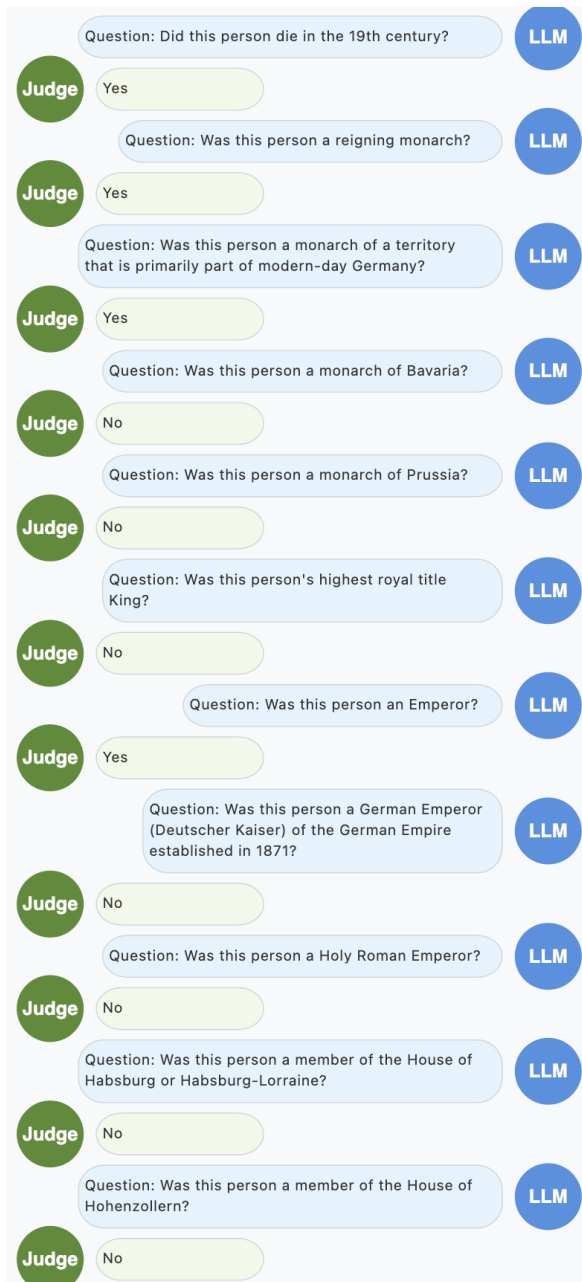


Figure 9: Part of a Gemini 3.1 Pro dialogue in the Alias game, where the hidden person is Napoleon. The model asks whether the subject died in the 19th century after already establishing that period through earlier temporal bounds. It then asks about entities such as the Holy Roman Empire and the House of Hohenzollern despite previously established dates, locations, and titles, illustrating redundant questioning and incomplete use of dialogue state.

for validation and opponent move generation. This dataset may include place names associated with politically sensitive regions, ethnic groups, historical conflicts, or other culturally sensitive contexts. Their inclusion reflects the geographic database and is not an expression of the authors’ political or cultural position.

For 20 Questions (Consistency), we use a fully synthetic dataset of morphologically plausible but non-existent words and synthetic hierarchical attributes. This dataset is procedurally generated and is designed specifically to avoid reliance on factual world knowledge.

For Situation Puzzles, 20 Features, and Alias with Personalities, datasets are constructed using a combination of LLM-based generation, automatic filtering, and, where applicable, human validation or external retrieval. Situation Puzzles are synthetically generated from predefined control parameters and filtered for clarity, realism, logical consistency, and semantic diversity. The 20 Features dataset is generated from a consensus vocabulary produced by multiple LLMs and enriched with model-generated attribute values. Alias with Personalities uses lists of notable public figures generated by multiple LLMs and retrieves biographical information from Wikipedia. We exclude entries with insufficient content and avoid using sensitive non-public information about living individuals.

All datasets are intended solely for evaluating language models in controlled multi-turn reasoning environments. No private user data was collected or used.

Artifact sources and licenses. English lexical validation uses NLTK 3.9.3, whose source code is distributed under Apache-2.0, and Princeton WordNet under the WordNet 3.0 license. Russian lexical validation uses the ruwordnet 0.0.6 Python wrapper, distributed under Apache-2.0, and the separately distributed, non-commercial-use Ru-WordNet thesaurus (Loukachevitch et al., 2016). The Word Chain locality files are adapted from the Geoapify World Cities Database (Geoapify, 2026), an OpenStreetMap-derived database distributed under ODbL; the artifact preserves OpenStreetMap attribution and documents our normalization and filtering. Alias biographies are adapted from Wikipedia and retain CC BY-SA 4.0/GFDL terms with per-file source attribution. The benchmark code is released under MIT, and only our original datasets and annotations are released

be understood as reflecting the coverage of the source resources, not the beliefs or preferences of the authors.

Second, several datasets are generated or assisted by large language models. Such models may reproduce biases, stereotypes, cultural assumptions, or artifacts from their training data. To mitigate this risk, we use multi-model generation, automatic quality checks, semantic filtering, and manual validation where appropriate. However, these procedures cannot guarantee complete removal of all biased, offensive, or culturally inappropriate content.

Third, the Alias with Personalities task involves public figures. We use publicly available biographical information retrieved from Wikipedia and exclude sensitive non-public information, particularly for living individuals. The task is intended to evaluate question-asking and identification strategies, not to make normative judgments about individuals. We do not include private persons or private user data.

Fourth, the benchmark includes bilingual English–Russian evaluation. Because cultural salience, lexical coverage, and data availability differ across languages, benchmark performance should not be interpreted as a direct measure of the intrinsic quality or cultural value of either language. Cross-lingual comparisons should be made cautiously and with attention to differences in lexical resources, corpora, and external knowledge coverage.

Finally, although the benchmark is intended for research evaluation, it may expose models to sensitive or ambiguous content. We encourage users of the benchmark to inspect datasets before deployment, apply additional filtering if required by their application context, and report results with appropriate caveats regarding possible dataset bias and limitations.

Cognitive Skill	Benchmark	Samples	Type	Turns	Task
Working Memory	WM (Gong et al., 2024)	1,200	Executive	Multi	Decide whether letter on each turn matches the letter shown 3 messages earlier.
Cognitive Flexibility	LLM-Cog.-Flex. (Kennedy and Nowak, 2024)	400	Executive	Multi	Infer a hidden categorization rule from a synthetic set of features, adapting when the rule silently switches (WCST + Letter-Number Test).
Causal Reasoning	CLADDER (Jin et al., 2024)	10,560	Executive	Single	Answer yes/no to formal causal queries from a set of well-defined formal rules and a situation description.
Logical Reasoning	LogiQA 2.0 (Liu et al., 2023)	1,572	Executive	Single	Pick correct answer from A/B/C/D on deductive-reasoning questions over short passages.
Common-sense Reasoning	WinoGrande (Sakaguchi et al., 2019)	1,267	Executive	Single	Resolve ambiguity in text passage by choosing which of two candidates fills the blank.
Planning	NATURAL PLAN (Zheng et al., 2024)	2,600	Executive	Single	Generate a trip itinerary or calendar schedule that satisfies constraints stated in natural language.
Emotional Intelligence	EQ-Bench (Paech, 2024)	171	Socio-Emo.	Single	Rate the intensity of four candidate emotions felt by a character in a short dialogue.
Pragmatics	LMPragmatics (Hu et al., 2023)	845	Socio-Emo.	Single	Pick the best interpretation of a situation across seven phenomena: deceit, indirect speech, irony, maxims, metaphor, humor, coherence.
Social Common-sense	SOCIAL IQA (Sap et al., 2019)	1,954	Socio-Emo.	Single	Pick a multiple-choice answer about a short social scenario (intentions, reactions, motivations, needs, consequences).
Theory of Mind	SimpleToM (Gu et al., 2026)	1,147	Socio-Emo.	Single	Decide whether a character is aware of a hidden fact (Simple-ToM mental-state subtask).
Attribution & Judgment	SimpleToM (Gu et al., 2026)	2,294	Socio-Emo.	Single	Predict a character’s likely behavior, or judge an action’s reasonableness, when the character lacks key information (Simple-ToM behavior + judgment subtasks).

Table 12: Cognitive benchmark suite.

Model	Executive (Core)		Executive (Higher-order)				Socio-emotional				
	WM	CogFlex	Causal	Logical	Common.	Plan	EQ	Pragm.	SocCom.	ToM	Attrib.
Qwen3 30B A3B Instruct 2507	0.69	0.36	0.57	0.75	0.72	0.26	0.80	0.80	0.78	0.88	0.44
Qwen3 235B A22B Instruct 2507	0.84	0.58	0.62	0.80	0.82	0.31	0.78	0.88	0.81	0.95	0.46
Qwen3.5 35B A3B	0.69	0.29	0.56	0.77	0.73	0.27	0.80	0.83	0.77	0.90	0.50
DeepSeek V3.1 Terminus	0.70	0.31	0.61	0.75	0.75	0.36	0.83	0.87	0.77	0.98	0.34
DeepSeek V4 Flash	0.67	0.35	0.62	0.77	0.74	0.37	0.82	0.89	0.77	0.89	0.40
DeepSeek V4 Pro	0.71	0.42	0.59	0.77	0.76	0.44	0.82	0.89	0.78	0.94	0.46
GPT 4.1	0.80	0.57	0.71	0.76	0.85	0.27	0.82	0.92	0.81	0.97	0.45
GPT 5.4 Nano	0.68	0.36	0.54	0.49	0.56	0.08	0.68	0.58	0.68	0.74	0.19
GPT 5.4 Mini	0.87	0.29	0.58	0.69	0.67	0.39	0.85	0.80	0.76	0.87	0.30
GPT 5.4	0.86	0.61	0.76	0.80	0.82	0.59	0.80	0.93	0.80	0.91	0.40
Gemini 3.1 Flash Lite	0.81	0.57	0.71	0.80	0.84	0.47	0.85	0.93	0.82	0.97	0.52
Gemini 3 Flash Preview	0.95	0.78	0.81	0.85	0.90	0.70	0.85	0.96	0.83	0.99	0.78

Table 13: Cognitive-benchmark accuracies per model (AO-only runs). Scores are in $[0, 1]$. Benchmarks are grouped per (Momentè et al., 2025): Executive Core (Working Memory, Cognitive Flexibility), Executive Higher-order (Causal Reasoning, Logical Reasoning, Common-sense Reasoning, Planning), and Socio-emotional (Emotional Intelligence, Pragmatics, Social Common-sense, Theory of Mind, Attribution & Judgment).

Model	Mode	English (EN)				Russian (RU)			
		FalsePosAttr	FalseNegAttr	WrongYes	WrongNo	FalsePosAttr	FalseNegAttr	WrongYes	WrongNo
Qwen3 30B A3B Instruct 2507	AO	68%	16%	0%	0%	74%	10%	0%	1%
	CoT	65%	9%	0%	0%	62%	7%	0%	0%
Qwen3 30B A3B Thinking 2507	Reason	0%	1%	0%	0%	2%	1%	0%	1%
Qwen3 235B A22B Instruct 2507	AO	72%	5%	0%	0%	65%	10%	0%	0%
	CoT	44%	4%	0%	0%	57%	6%	0%	0%
Qwen3 235B A22B Thinking 2507	Reason	2%	1%	0%	0%	3%	1%	0%	0%
Qwen3.5 35B A3B	AO	70%	16%	0%	0%	73%	20%	0%	0%
	Reason	0%	1%	0%	0%	2%	3%	0%	0%
DeepSeek V3.1 Terminus	AO	40%	2%	0%	0%	42%	1%	0%	0%
	Reason	1%	0%	0%	0%	8%	0%	0%	0%
DeepSeek V4 Flash	AO	40%	15%	0%	0%	66%	5%	0%	0%
	Reason	1%	0%	0%	0%	15%	3%	0%	0%
DeepSeek V4 Pro	AO	13%	3%	0%	0%	39%	7%	0%	0%
	Reason	0%	0%	0%	0%	8%	1%	1%	0%
GPT OSS 120B	Reason	13%	1%	1%	0%	1%	6%	0%	1%
GPT 4.1	AO	51%	1%	0%	0%	48%	6%	0%	0%
	CoT	30%	10%	0%	0%	43%	15%	0%	0%
GPT 5.4 Nano	AO	84%	10%	0%	0%	88%	10%	0%	0%
	Reason	41%	13%	0%	1%	32%	9%	0%	3%
GPT 5.4 Mini	AO	67%	8%	0%	0%	69%	18%	0%	0%
	Reason	20%	2%	0%	0%	5%	2%	0%	0%
GPT 5.4	AO	3%	2%	0%	0%	9%	4%	0%	0%
	Reason	2%	0%	0%	0%	4%	0%	0%	0%
Gemini 3.1 Flash Lite	AO	3%	5%	0%	0%	11%	2%	0%	0%
	Reason	1%	0%	0%	0%	1%	0%	0%	0%
Gemini 3 Flash Preview	AO	16%	1%	0%	0%	7%	0%	0%	0%
	Reason	8%	1%	0%	0%	0%	0%	0%	0%
Gemini 3.1 Pro Preview	Reason	1%	1%	0%	0%	1%	0%	0%	0%
Human	AO	-	-	-	-	7%	5%	0%	0%

Table 14: Failure-mode breakdown for 20 Questions. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

Model	Mode	English (EN)						Russian (RU)					
		WrongLen	PatViol	NonWord	RepLetter	FmtFail	NoWordGuess	WrongLen	PatViol	NonWord	RepLetter	FmtFail	NoWordGuess
Qwen3 30B A3B Instruct 2507	AO	8%	1%	2%	73%	0%	34%	22%	8%	8%	76%	1%	18%
	CoT	42%	16%	14%	11%	16%	0%	59%	31%	49%	37%	48%	0%
Qwen3 30B A3B Thinking 2507	Reason	18%	19%	27%	5%	11%	0%	21%	13%	31%	25%	9%	0%
Qwen3 235B A22B Instruct 2507	AO	28%	11%	8%	8%	2%	0%	24%	12%	23%	37%	4%	0%
	CoT	12%	13%	0%	3%	0%	0%	8%	1%	8%	3%	0%	0%
Qwen3 235B A22B Thinking 2507	Reason	0%	0%	0%	0%	0%	0%	2%	2%	8%	2%	0%	0%
Qwen3.5 35B A3B	AO	0%	0%	1%	34%	1%	1%	14%	4%	3%	77%	2%	15%
	Reason	31%	0%	6%	4%	0%	0%	12%	4%	8%	28%	2%	0%
DeepSeek V3.1 Terminus	AO	11%	5%	5%	5%	18%	0%	14%	9%	23%	10%	16%	0%
	Reason	0%	0%	0%	1%	0%	0%	0%	1%	5%	2%	1%	0%
DeepSeek V4 Flash	AO	9%	4%	11%	1%	1%	0%	16%	2%	15%	25%	10%	2%
	Reason	1%	2%	7%	4%	1%	0%	4%	5%	10%	10%	11%	0%
DeepSeek V4 Pro	AO	1%	1%	0%	1%	9%	0%	11%	4%	10%	4%	6%	0%
	Reason	1%	3%	1%	0%	9%	0%	9%	7%	16%	2%	6%	0%
GPT OSS 120B	Reason	7%	4%	22%	7%	32%	0%	13%	3%	41%	32%	23%	0%
GPT 4.1	AO	11%	4%	5%	1%	32%	0%	10%	2%	4%	4%	33%	2%
	CoT	21%	11%	0%	3%	0%	0%	13%	2%	1%	12%	2%	0%
GPT 5.4 Nano	AO	59%	34%	11%	38%	35%	2%	69%	39%	35%	65%	85%	1%
	Reason	9%	4%	17%	2%	8%	0%	3%	3%	46%	2%	16%	0%
GPT 5.4 Mini	AO	26%	15%	12%	11%	38%	1%	57%	53%	48%	21%	45%	0%
	Reason	1%	0%	11%	0%	8%	0%	1%	5%	35%	3%	3%	0%
GPT 5.4	AO	2%	2%	1%	0%	2%	0%	13%	19%	25%	2%	14%	0%
	Reason	0%	0%	2%	0%	0%	0%	0%	0%	15%	0%	0%	0%
Gemini 3.1 Flash Lite	AO	9%	20%	13%	1%	10%	0%	39%	37%	40%	4%	7%	0%
	Reason	1%	4%	6%	0%	0%	0%	4%	4%	20%	0%	7%	0%
Gemini 3 Flash Preview	AO	9%	12%	5%	1%	2%	0%	4%	5%	6%	1%	18%	0%
	Reason	0%	0%	0%	0%	2%	0%	0%	0%	0%	0%	1%	0%
Gemini 3.1 Pro Preview	Reason	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Human	AO	-	-	-	-	-	-	1%	0%	2%	9%	1%	0%

Table 15: Failure-mode breakdown for Wheel of Fortune. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

Model	Mode	English (EN)			Russian (RU)		
		WordRepeat	InvalidWord	WrongLen	WordRepeat	InvalidWord	WrongLen
Qwen3 30B A3B Instruct 2507	AO	92%	83%	62%	96%	99%	89%
	CoT	35%	56%	23%	67%	92%	69%
Qwen3 30B A3B Thinking 2507	Reason	17%	51%	0%	46%	85%	16%
Qwen3 235B A22B Instruct 2507	AO	57%	93%	57%	73%	100%	80%
	CoT	20%	51%	10%	40%	79%	43%
Qwen3 235B A22B Thinking 2507	Reason	5%	33%	0%	29%	70%	12%
Qwen3.5 35B A3B	AO	84%	94%	66%	88%	95%	98%
	Reason	7%	35%	0%	20%	79%	27%
DeepSeek V3.1 Terminus	AO	22%	70%	20%	89%	85%	74%
	Reason	3%	40%	1%	16%	51%	8%
DeepSeek V4 Flash	AO	53%	93%	38%	60%	91%	67%
	Reason	12%	34%	0%	25%	56%	8%
DeepSeek V4 Pro	AO	23%	76%	16%	37%	88%	57%
	Reason	1%	36%	1%	8%	37%	16%
GPT OSS 120B	Reason	9%	46%	0%	42%	68%	13%
GPT 4.1	AO	44%	76%	36%	60%	97%	63%
	CoT	20%	61%	13%	56%	96%	48%
GPT 5.4 Nano	AO	86%	97%	47%	89%	95%	73%
	Reason	16%	40%	0%	36%	77%	9%
GPT 5.4 Mini	AO	28%	92%	61%	36%	97%	62%
	Reason	10%	24%	0%	14%	47%	1%
GPT 5.4	AO	26%	75%	28%	29%	93%	60%
	Reason	1%	11%	0%	0%	27%	2%
Gemini 3.1 Flash Lite	AO	41%	84%	17%	67%	97%	35%
	Reason	8%	41%	0%	30%	75%	13%
Gemini 3 Flash Preview	AO	25%	70%	16%	40%	83%	48%
	Reason	3%	14%	1%	2%	17%	1%
Gemini 3.1 Pro Preview	Reason	0%	4%	0%	0%	10%	0%
Human	AO	-	-	-	0%	33%	5%

Table 16: Failure-mode breakdown for Wordle. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

Model	Mode	English (EN)	Russian (RU)
		NotQuiteRight	NotQuiteRight
Qwen3 30B A3B Instruct 2507	AO	46%	34%
	CoT	49%	77%
Qwen3 30B A3B Thinking 2507	Reason	51%	55%
Qwen3 235B A22B Instruct 2507	AO	24%	39%
	CoT	40%	49%
Qwen3 235B A22B Thinking 2507	Reason	30%	38%
Qwen3.5 35B A3B	AO	24%	46%
	Reason	29%	35%
DeepSeek V3.1 Terminus	AO	19%	26%
	Reason	32%	26%
DeepSeek V4 Flash	AO	18%	34%
	Reason	22%	25%
DeepSeek V4 Pro	AO	18%	27%
	Reason	26%	24%
GPT OSS 120B	Reason	62%	57%
GPT 4.1	AO	15%	29%
	CoT	18%	45%
GPT 5.4 Nano	AO	43%	48%
	Reason	34%	33%
GPT 5.4 Mini	AO	30%	37%
	Reason	26%	25%
GPT 5.4	AO	17%	22%
	Reason	17%	19%
Gemini 3.1 Flash Lite	AO	18%	19%
	Reason	22%	26%
Gemini 3 Flash Preview	AO	16%	15%
	Reason	22%	12%
Gemini 3.1 Pro Preview	Reason	12%	16%
Human	AO	-	40%

Table 17: Failure-mode breakdown for Situation Puzzles. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

Model	Mode	English (EN)	Russian (RU)
		RepAnswer	RepAnswer
Qwen3 30B A3B Instruct 2507	AO	2%	7%
	CoT	40%	33%
Qwen3 30B A3B Thinking 2507	Reason	14%	10%
Qwen3 235B A22B Instruct 2507	AO	1%	4%
	CoT	39%	13%
Qwen3 235B A22B Thinking 2507	Reason	14%	10%
Qwen3.5 35B A3B	AO	32%	37%
	Reason	2%	2%
DeepSeek V3.1 Terminus	AO	3%	16%
	Reason	4%	2%
DeepSeek V4 Flash	AO	12%	4%
	Reason	5%	19%
DeepSeek V4 Pro	AO	1%	5%
	Reason	29%	35%
GPT OSS 120B	Reason	3%	11%
GPT 4.1	AO	2%	2%
	CoT	1%	0%
GPT 5.4 Nano	AO	5%	12%
	Reason	30%	1%
GPT 5.4 Mini	AO	17%	11%
	Reason	7%	4%
GPT 5.4	AO	1%	3%
	Reason	0%	0%
Gemini 3.1 Flash Lite	AO	9%	11%
	Reason	0%	0%
Gemini 3 Flash Preview	AO	0%	2%
	Reason	0%	0%
Gemini 3.1 Pro Preview	Reason	1%	0%
Human	AO	-	7%

Table 18: Failure-mode breakdown for 20 Features. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

Model	Mode	English (EN)	Russian (RU)
		GuessGtOnce	GuessGtOnce
Qwen3 30B A3B Instruct 2507	AO	2%	4%
	CoT	40%	31%
Qwen3 30B A3B Thinking 2507	Reason	5%	13%
Qwen3 235B A22B Instruct 2507	AO	0%	2%
	CoT	39%	11%
Qwen3 235B A22B Thinking 2507	Reason	0%	3%
Qwen3.5 35B A3B	AO	1%	1%
	Reason	1%	1%
DeepSeek V3.1 Terminus	AO	2%	16%
	Reason	2%	1%
DeepSeek V4 Flash	AO	3%	10%
	Reason	3%	6%
DeepSeek V4 Pro	AO	0%	0%
	Reason	9%	6%
GPT OSS 120B	Reason	2%	7%
GPT 4.1	AO	2%	1%
	CoT	1%	0%
GPT 5.4 Nano	AO	1%	4%
	Reason	0%	0%
GPT 5.4 Mini	AO	12%	8%
	Reason	4%	2%
GPT 5.4	AO	1%	0%
	Reason	0%	0%
Gemini 3.1 Flash Lite	AO	2%	11%
	Reason	0%	0%
Gemini 3 Flash Preview	AO	0%	2%
	Reason	0%	0%
Gemini 3.1 Pro Preview	Reason	1%	0%
Human	AO	-	50%

Table 19: Failure-mode breakdown for Alias. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

Model	Mode	English (EN)			Russian (RU)		
		RepCity	WrongLetter	NotFound	RepCity	WrongLetter	NotFound
Qwen3 30B A3B Instruct 2507	AO	26%	59%	15%	17%	58%	25%
	CoT	35%	52%	9%	63%	32%	5%
Qwen3 30B A3B Thinking 2507	Reason	10%	13%	7%	22%	47%	1%
Qwen3 235B A22B Instruct 2507	AO	43%	51%	6%	14%	71%	15%
	CoT	62%	14%	8%	62%	22%	14%
Qwen3 235B A22B Thinking 2507	Reason	0%	5%	0%	4%	28%	2%
Qwen3.5 35B A3B	AO	2%	95%	3%	3%	88%	9%
	Reason	1%	28%	0%	5%	41%	0%
DeepSeek V3.1 Terminus	AO	79%	18%	2%	10%	79%	7%
	Reason	8%	3%	0%	3%	47%	11%
DeepSeek V4 Flash	AO	57%	13%	16%	18%	46%	21%
	Reason	70%	3%	7%	45%	26%	11%
DeepSeek V4 Pro	AO	46%	8%	9%	9%	74%	6%
	Reason	36%	5%	4%	22%	39%	12%
GPT OSS 120B	Reason	3%	22%	3%	10%	12%	14%
GPT 4.1	AO	78%	8%	8%	14%	58%	23%
	CoT	11%	1%	10%	18%	54%	8%
GPT 5.4 Nano	AO	0%	78%	22%	5%	77%	18%
	Reason	5%	14%	6%	9%	29%	4%
GPT 5.4 Mini	AO	69%	19%	9%	34%	54%	10%
	Reason	8%	49%	4%	6%	78%	0%
GPT 5.4	AO	64%	15%	2%	33%	38%	5%
	Reason	0%	9%	2%	0%	6%	0%
Gemini 3.1 Flash Lite	AO	68%	6%	10%	2%	86%	5%
	Reason	0%	3%	1%	0%	26%	4%
Gemini 3 Flash Preview	AO	37%	15%	7%	3%	46%	6%
	Reason	0%	0%	3%	0%	0%	0%
Gemini 3.1 Pro Preview	Reason	0%	0%	1%	0%	1%	1%
Human	AO	-	-	-	12%	7%	33%

Table 20: Failure-mode breakdown for Word Chain. Each cell is the percent of analyzed games in which the failure was detected at least once. AO = answer-only; CoT = chain-of-thought prompt; Reason = internal-reasoning regime.

L Prompts

This appendix reports every player prompt used in the answer-only (AO) and chain-of-thought (CoT) setups in both benchmark languages. Values that vary by episode or turn are shown as placeholders in braces. For each of the three LLM-judged games, one judge box contains all English and Russian templates used by that judge.

L.1 Player Prompts

L.1.1 Wordle

Wordle – AO – English

[user]

You are playing Wordle. Your task is to guess the
→ hidden 5-letter English word in as few
→ attempts as possible.

Rules:

1. The hidden word is a singular common noun.
2. On each turn, provide exactly one 5-letter word.
3. After each guess, you receive letter feedback:
 - Green: the letter is in the correct position.
 - Yellow: the letter is in the word but in a
→ different position.
 - Gray: the letter is not in the word.
4. Use previous feedback to make a better next
→ guess.

Example:

Hidden word: "crane".
Player guesses: "cared".

Feedback:

- "c" — green letter, because it is in the hidden
→ word and in the correct position.
- "a" — yellow letter, because it is in the hidden
→ word but in a different position.
- "r" — yellow letter, because it is in the hidden
→ word but in a different position.
- "e" — yellow letter, because it is in the hidden
→ word but in a different position.
- "d" — gray letter, because this letter is not in
→ the hidden word.

You MUST use the exact answer format:
Answer: <5-letter word>

IMPORTANT:

- Provide only one word.
- Do not add anything after the final word.

Let's start the game. Give your first word.

Wordle – AO – Russian

[user]

Ты играешь в Wordle. Твоя задача — угадать
→ скрытое русское слово из 5 букв за
→ минимальное количество попыток.

Правила игры:

1. Загаданное слово — существительное в
→ единственном числе, именительном
→ падеже.
2. Вместо буквы "ё" используй букву "е".
3. На каждом ходу ты называешь одно слово из
→ 5 букв.
4. После каждой попытки ты получаешь разбор
→ букв:
 - Зеленый: буква стоит на правильном месте.
 - Желтый: буква есть в слове, но стоит в
→ другом месте.
 - Серый: буквы нет в слове.
5. Используй предыдущую обратную связь,
→ чтобы сузить гипотезу.

Пример:

Загадано слово: "книга".

Игрок называет слово: "канат".

Разбор:

- "к" — зеленая буква, потому что она есть в
→ загаданном слове и стоит на правильном
→ месте.
- "а" — желтая буква, потому что она есть в
→ загаданном слове, но стоит в другом месте.
- "н" — желтая буква, потому что она есть в
→ загаданном слове, но стоит в другом месте.
- "и" — желтая буква, потому что она есть в
→ загаданном слове, но стоит в другом месте.
- "т" — серая буква, потому что такой буквы
→ нет в загаданном слове.

ОБЯЗАТЕЛЬНО используй точный формат
→ ответа:

Ответ: <слово из 5 букв>

ВАЖНО:

- Пиши только одно слово.
- Не добавляй пояснений после итогового слова.

Начинаем игру. Ваш первый вариант слова.

Wordle – CoT – English

[user]

You are playing Wordle. Your task is to guess the
→ hidden 5-letter English word in as few
→ attempts as possible.

Rules:

1. The hidden word is a singular common noun.
2. On each turn, provide exactly one 5-letter word.
3. After each guess, you receive letter feedback:
 - Green: the letter is in the correct position.
 - Yellow: the letter is in the word but in a
→ different position.
 - Gray: the letter is not in the word.
4. Use previous feedback to make a better next
→ guess.

Example:

Hidden word: "crane".
Player guesses: "cared".

Feedback:

- "c" — green letter, because it is in the hidden word and in the correct position.
- "a" — yellow letter, because it is in the hidden word but in a different position.
- "r" — yellow letter, because it is in the hidden word but in a different position.
- "e" — yellow letter, because it is in the hidden word but in a different position.
- "d" — gray letter, because this letter is not in the hidden word.

Briefly explain your hypothesis in reasoning, then
→ answer. The required format:

Reasoning: <your thoughts>
Answer: <5-letter word>

You MUST use the exact answer format:
Answer: <5-letter word>

IMPORTANT:
- Provide only one word.
- Do not add anything after the final word.

Let's start the game. Give your first word.

Wordle – CoT – Russian

[user]

Ты играешь в Wordle. Твоя задача — угадать
→ скрытое русское слово из 5 букв за
→ минимальное количество попыток.

Правила игры:

1. Загаданное слово — существительное в единственном числе, именительном падеже.
2. Вместо буквы "ё" используй букву "е".
3. На каждом ходу ты называешь одно слово из 5 букв.
4. После каждой попытки ты получаешь разбор
→ букв:
 - Зеленый: буква стоит на правильном месте.
 - Желтый: буква есть в слове, но стоит в другом месте.
 - Серый: буквы нет в слове.
5. Используй предыдущую обратную связь,
→ чтобы сузить гипотезу.

Пример:

Загадано слово: "книга".

Игрок называет слово: "канат".

Разбор:

- "к" — зеленая буква, потому что она есть в загаданном слове и стоит на правильном месте.
- "а" — желтая буква, потому что она есть в загаданном слове, но стоит в другом месте.
- "н" — желтая буква, потому что она есть в загаданном слове, но стоит в другом месте.
- "а" — желтая буква, потому что она есть в загаданном слове, но стоит в другом месте.
- "т" — серая буква, потому что такой буквы нет в загаданном слове.

Сначала кратко объясни гипотезу в
→ рассуждениях, затем напиши ответ.
→ Необходимый формат:

Рассуждение: <ваши мысли>

Ответ: <слово из 5 букв>

ОБЯЗАТЕЛЬНО используй точный формат

→ ответа:

Ответ: <слово из 5 букв>

ВАЖНО:

- Пиши только одно слово.
- Не добавляй пояснений после итогового слова.

Начинаем игру. Ваш первый вариант слова.

L.1.2 Word Chain

Word Chain – AO – English

[user]

You are playing the "Cities" game. This is a
→ classic word game where players take turns
→ naming cities.

Rules:

1. I name a city.
2. You must name a city that starts with the last letter of my city.
3. You cannot repeat cities that have already been used in this game.
4. You do not need to verify my cities. They always come from the database.
5. You should only name existing cities and use their current names. For instance,
→ "Novgorod" is wrong because nowadays it is "Veliky Novgorod".
6. Use full English city names in ASCII.

Examples:

- Me: "London" (ends with "n")
- You: "New York" (starts with "n", ends with "k",
→ so the next letter is "k")
- Me: "Kyoto" (starts with "k", ends with "o")
- You: "Oslo" (starts with "o")

You MUST use the exact FINAL answer format:

Answer: <full city name>

IMPORTANT:

- Your reply must start with "Answer:".
- Do not add anything after the city name.
- Do not ask questions or comment on my move.

Let's start. My opening city is "{opening city}".
→ Your move.

Word Chain – AO – Russian

[user]

Ты играешь в игру "Города". Это классическая
→ словесная игра, где участники по очереди
→ называют города.

Правила игры:

1. Я называю город.
2. Ты должен назвать город, который
→ начинается на последнюю допустимую
→ букву моего города.
3. Нельзя повторять города, которые уже были
→ названы в этой игре.

4. Если город заканчивается на букву "ь", "ъ"
→ или "ы", нужно использовать
→ предпоследнюю букву.
5. Я называю только корректные города из
→ базы данных, их проверять не нужно.
6. Нужно называть только ныне существующие
→ города и использовать их актуальные
→ названия. Например, называть "Новгород"
→ неправильно, потому что сейчас это
→ "Великий Новгород".
7. Нужно писать полные названия городов на
→ русском языке.

Примеры:

- Я: "Москва" (заканчивается на "а")
- Ты: "Астрахань" (начинается на "а",
→ заканчивается на "ь", значит следующая
→ буква - "н")
- Я: "Новосибирск" (начинается на "н",
→ заканчивается на "к")
- Ты: "Казань" (начинается на "к")

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа:

Ответ: <полное название города>

ВАЖНО:

- Окончательный ответ должен начинаться со
→ слова "Ответ:".
- Не добавляй ничего после названия города.
- Не задавай вопросов и не комментируй мой
→ ход.

Начинаем игру. Я начинаю с города "{opening
→ city}". Ваш ход.

Word Chain – CoT – English

[user]

You are playing the "Cities" game. This is a
→ classic word game where players take turns
→ naming cities.

Rules:

1. I name a city.
2. You must name a city that starts with the last
→ letter of my city.
3. You cannot repeat cities that have already been
→ used in this game.
4. You do not need to verify my cities. They
→ always come from the database.
5. You should only name existing cities and use
→ their current names. For instance,
→ "Novgorod" is wrong because nowadays it is
→ "Veliky Novgorod".
6. Use full English city names in ASCII.

Examples:

- Me: "London" (ends with "n")
- You: "New York" (starts with "n", ends with "k",
→ so the next letter is "k")
- Me: "Kyoto" (starts with "k", ends with "o")
- You: "Oslo" (starts with "o")

Briefly explain your choice in reasoning, then
→ answer. The required format:

Reasoning: <your thoughts>

Answer: <full city name>

You **MUST** use the exact FINAL answer format:

Answer: <full city name>

IMPORTANT:

- Your reply must start with "Answer:".
- Do not add anything after the city name.
- Do not ask questions or comment on my move.

Let's start. My opening city is "{opening city}".

→ Your move.

Word Chain – CoT – Russian

[user]

Ты играешь в игру "Города". Это классическая
→ словесная игра, где участники по очереди
→ называют города.

Правила игры:

1. Я называю город.
2. Ты должен назвать город, который
→ начинается на последнюю допустимую
→ букву моего города.
3. Нельзя повторять города, которые уже были
→ названы в этой игре.
4. Если город заканчивается на букву "ь", "ъ"
→ или "ы", нужно использовать
→ предпоследнюю букву.
5. Я называю только корректные города из
→ базы данных, их проверять не нужно.
6. Нужно называть только ныне существующие
→ города и использовать их актуальные
→ названия. Например, называть "Новгород"
→ неправильно, потому что сейчас это
→ "Великий Новгород".
7. Нужно писать полные названия городов на
→ русском языке.

Примеры:

- Я: "Москва" (заканчивается на "а")
- Ты: "Астрахань" (начинается на "а",
→ заканчивается на "ь", значит следующая
→ буква - "н")
- Я: "Новосибирск" (начинается на "н",
→ заканчивается на "к")
- Ты: "Казань" (начинается на "к")

Сначала кратко объясни выбор, а затем

→ напиши ответ в формате:

Рассуждение: <твои мысли>

Ответ: <полное название города>

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа:

Ответ: <полное название города>

ВАЖНО:

- Окончательный ответ должен начинаться со
→ слова "Ответ:".
- Не добавляй ничего после названия города.
- Не задавай вопросов и не комментируй мой
→ ход.

Начинаем игру. Я начинаю с города "{opening
→ city}". Ваш ход.

L.1.3 Situation Puzzles

Situation Puzzles – AO – English

[user]

You are playing a lateral thinking puzzle. You
→ will see a brief setup of a strange situation,
→ but you do not know the real solution.

Your task:

1. Ask the host clarifying questions that can be
→ answered with "yes", "no", or "it doesn't
→ matter".
2. Use the answers to reconstruct what happened.
3. When you are confident, provide the final
→ solution.
4. On each turn, do exactly one thing: ask one
→ question or give one final answer.

Rules:

- Ask specific and useful questions.
- Do not combine multiple independent questions
→ in one turn.
- Use the previous answers consistently.
- Do not invent supernatural explanations unless
→ the setup clearly requires them.

ALWAYS use the exact format for the final
→ answer or question:

Answer: <complete solution> or Question: <your
→ question>

IMPORTANT:

- The final answer must begin with the word
→ "Answer:" and the question with the word
→ "Question:".
- Do not add anything after the answer or
→ question.

Here is the setup of the lateral thinking puzzle:

{puzzle setup}

Make your first move: either one clarifying
→ question or a final solution.

Moves left until the end of the game: {moves left}

Situation Puzzles – AO – Russian

[user]

Ты играешь в данетку. Тебе будет дана завязка
→ странной истории, но истинная разгадка
→ тебе неизвестна.

Твоя задача:

1. Задавать ведущему уточняющие вопросы, на
→ которые можно ответить "да", "нет" или
→ "неважно".
2. На основе ответов постепенно восстановить
→ полную картину произошедшего.
3. Когда будешь уверен в решении, выдать
→ финальную разгадку.
4. За один ход можно сделать только одно
→ действие: задать один вопрос или
→ предложить финальный ответ.

Правила:

- Задавай конкретные и полезные вопросы.
- Не объединяй несколько независимых
→ вопросов в один ход.
- Учитывай историю предыдущих ответов.
- Не придумывай сверхъестественные
→ объяснения без явных оснований.

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа или вопроса:
Ответ: <полное решение загадки> или Вопрос:
→ <твой вопрос>

ВАЖНО:

- Окончательный ответ должен начинаться со
→ слова "Ответ:", а вопрос со слова
→ "Вопрос:".
- Не добавляй ничего после ответа или вопроса.

Вот завязка данетки:

{puzzle setup}

Сформулируй первый ход: либо один
→ уточняющий вопрос, либо финальную
→ разгадку.

Осталось ходов до конца игры: {moves left}

Situation Puzzles – CoT – English

[user]

You are playing a lateral thinking puzzle. You
→ will see a brief setup of a strange situation,
→ but you do not know the real solution.

Your task:

1. Ask the host clarifying questions that can be
→ answered with "yes", "no", or "it doesn't
→ matter".
2. Use the answers to reconstruct what happened.
3. When you are confident, provide the final
→ solution.
4. On each turn, do exactly one thing: ask one
→ question or give one final answer.

Rules:

- Ask specific and useful questions.
- Do not combine multiple independent questions
→ in one turn.
- Use the previous answers consistently.
- Do not invent supernatural explanations unless
→ the setup clearly requires them.

Reason briefly first, then choose exactly one
→ action in one of these formats:

Reasoning: <your thoughts>

Question: <one specific question>

or

Reasoning: <your thoughts>

Answer: <your solution>

ALWAYS use the exact format for the final

→ answer or question:

Answer: <complete solution> or Question: <your
→ question>

IMPORTANT:

- The final answer must begin with the word
→ "Answer:" and the question with the word
→ "Question:".
- Do not add anything after the answer or
→ question.

Here is the setup of the lateral thinking puzzle:

{puzzle setup}

Make your first move with reasoning: either one
→ clarifying question or a final solution.

Moves left until the end of the game: {moves left}

Situation Puzzles – CoT – Russian

[user]

Ты играешь в данетки. Тебе будет дана завязка
→ странной истории, но истинная разгадка
→ тебе неизвестна.

Твоя задача:

1. Задавать ведущему уточняющие вопросы, на
→ которые можно ответить "да", "нет" или
→ "неважно".
2. На основе ответов постепенно восстановить
→ полную картину произошедшего.
3. Когда будешь уверен в решении, выдать
→ финальную разгадку.
4. За один ход можно сделать только одно
→ действие: задать один вопрос или
→ предложить финальный ответ.

Правила:

- Задавай конкретные и полезные вопросы.
- Не объединяй несколько независимых
→ вопросов в один ход.
- Учитывай историю предыдущих ответов.
- Не придумывай сверхъестественные
→ объяснения без явных оснований.

Сначала кратко рассуждай, а затем выбери
→ ровно одно действие в одном из форматов:

Рассуждение: <ваши мысли>

Вопрос: <один конкретный вопрос>

или

Рассуждение: <ваши мысли>

Ответ: <ваша разгадка>

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа или вопроса:

Ответ: <полное решение загадки> или Вопрос:

→ <твой вопрос>

ВАЖНО:

- Окончательный ответ должен начинаться со
→ слова "Ответ:", а вопрос со слова
→ "Вопрос:".
- Не добавляй ничего после ответа или вопроса.

Вот завязка данетки:

{puzzle setup}

Сформулируй первый ход с рассуждениями:

→ либо один уточняющий вопрос, либо
→ финальную разгадку.

Осталось ходов до конца игры: {moves left}

L.1.4 20 Questions

20 Questions – AO – English

[user]

Let's play the 20 Questions game. I want you to
→ use the secret word that I have assigned to
→ you.

Here is the list of all possible words:

{word pool}

Here are the attributes of each word:

{word attributes}

I will ask you questions, and you should answer

- only "yes" or "no" depending on whether the
- attribute I am asking about is present in the
- secret word's attributes. You should answer
- honestly.

Do not name the word itself until it is explicitly
→ asked for.

When I think I know the word, I will ask you if

- that word is the secret word. You should
- answer "yes" or "no" depending on whether I
- guessed correctly.

Don't write any reasoning, give the final answer

→ immediately in the format

Answer: <your answer>.

You **MUST** use the exact final answer format!

Answer: yes

or

Answer: no

IMPORTANT:

- The final answer must start with the word
→ "Answer:".
- Do not add anything after "yes" or "no".
- Do not ask questions or comment on my move.

****YOUR ASSIGNED SECRET WORD:****

→ {secret word}

Ready to start? Answer "Answer: yes".

20 Questions – AO – Russian

[user]

Давайте поиграем в игру из 20 вопросов. Я
→ хочу, чтобы вы использовали секретное
→ слово, которое я вам назначил.

Вот список всех возможных слов:

{word pool}

Вот атрибуты каждого слова:

{word attributes}

Я буду задавать вам вопросы, и вы должны
→ отвечать только «да» или «нет», в
→ зависимости от того, есть ли атрибут, о
→ котором я спрашиваю, у секретного слова.
→ Вы должны отвечать честно.
Не называйте само слово, пока оно не будет
→ запрошено явно.

Когда я пойму, что знаю слово, я спрошу вас,
→ является ли это слово секретным словом.
→ Вы должны ответить «да» или «нет», в
→ зависимости от того, правильно ли я
→ угадал.

Не пиши никаких рассуждений, дай сразу же
→ окончательный ответ в формате
Ответ: <ваш ответ>.

ОБЯЗАТЕЛЬНО используйте точный формат
→ окончательного ответа!

Ответ: да
или
Ответ: нет

ВАЖНО:
- Окончательный ответ должен начинаться со
→ слова "Ответ:".
- Не добавляй ничего после "да" или "нет".
- Не задавай вопросов и не комментируй мой
→ ход.

****ВАШЕ НАЗНАЧЕННОЕ СЕКРЕТНОЕ**
→ **СЛОВО:**** {secret word}

Готовы начать? Ответьте "Ответ: да".

20 Questions – CoT – English

[user]
Let's play the 20 Questions game. I want you to
→ use the secret word that I have assigned to
→ you.

Here is the list of all possible words:

{word pool}

Here are the attributes of each word:

{word attributes}

I will ask you questions, and you should answer
→ only "yes" or "no" depending on whether the
→ attribute I am asking about is present in the
→ secret word's attributes. You should answer
→ honestly.

Do not name the word itself until it is explicitly
→ asked for.

When I think I know the word, I will ask you if
→ that word is the secret word. You should
→ answer "yes" or "no" depending on whether I
→ guessed correctly.

Explain your reasoning before answering in the
→ format:

Reasoning: <your thoughts>

Answer: <your answer>

You **MUST** use the exact final answer format!

Answer: yes

or

Answer: no

IMPORTANT:

- The final answer must start with the word
→ "Answer:".
- Do not add anything after "yes" or "no".
- Do not ask questions or comment on my move.

****YOUR ASSIGNED SECRET WORD:****

→ {secret word}

Ready to start? Answer "Answer: yes".

20 Questions – CoT – Russian

[user]
Давайте поиграем в игру из 20 вопросов. Я
→ хочу, чтобы вы использовали секретное
→ слово, которое я вам назначил.

Вот список всех возможных слов:

{word pool}

Вот атрибуты каждого слова:

{word attributes}

Я буду задавать вам вопросы, и вы должны
→ отвечать только «да» или «нет», в
→ зависимости от того, есть ли атрибут, о
→ котором я спрашиваю, у секретного слова.
→ Вы должны отвечать честно.
Не называйте само слово, пока оно не будет
→ запрошено явно.

Когда я пойму, что знаю слово, я спрошу вас,
→ является ли это слово секретным словом.
→ Вы должны ответить «да» или «нет», в
→ зависимости от того, правильно ли я
→ угадал.

Объясни свои рассуждения перед ответом в
→ формате:

Рассуждение: <ваши мысли>

Ответ: <ваш ответ>

ОБЯЗАТЕЛЬНО используйте точный формат
→ окончательного ответа!

Ответ: да
или
Ответ: нет

ВАЖНО:
- Окончательный ответ должен начинаться со
→ слова "Ответ:".
- Не добавляй ничего после "да" или "нет".

- Не задавай вопросов и не комментируй мой
→ ход.

****ВАШЕ НАЗНАЧЕННОЕ СЕКРЕТНОЕ**
→ СЛОВО:** {secret word}

Готовы начать? Ответьте "Ответ: да".

L.1.5 Wheel of Fortune

Wheel of Fortune – AO – English

[user]
Let's play "Wheel of Fortune".

In this game, you need to guess a hidden word.
→ You can guess letters one at a time or try to
→ guess the whole word. When you guess a
→ letter, if it is in the word, it will be revealed.
→ If the letter appears multiple times in the
→ word, all occurrences will be revealed. You
→ win either by guessing the whole word or by
→ guessing all its letters. You lose if you run
→ out of attempts without guessing the word.

****Rules:****
- You have {maximum turns} attempts

Don't write any reasoning, give the final answer
→ immediately in the format
Answer: <your answer>.

You MUST use the exact final answer format!

Example of guessing a letter:
Answer: E

Example of guessing a word:
Answer: WORD: EXAMPLE

IMPORTANT:
- The final answer must start with the word
→ "Answer:".
- Do not add anything after the guessed letter or
→ word.
- Do not ask questions or comment on my move.

****Current state:****
Word: {revealed word}
Attempts used: {turns used}/{maximum turns}

Let's start the game! Your move.

Wheel of Fortune – AO – Russian

[user]
Давай сыграем в "Поле чудес".

В этой игре от вас требуется угадать скрытое
→ слово. Вы можете угадывать буквы по
→ одной или пытаться угадать слово
→ целиком. Когда вы угадываете букву, если
→ она есть в слове, она открывается. Если
→ буква присутствует в слове несколько раз,
→ откроются все её вхождения. Вы
→ выигрываете либо угадав слово целиком,
→ либо угадав все его буквы. Вы
→ проигрываете, если исчерпаете все свои
→ попытки, не угадав слово.

****Правила:****
- У вас есть {maximum turns} попыток
- ****Эквивалентные буквы****: Следующие буквы
→ считаются взаимозаменяемыми:
- И и Й (можно использовать любую)
- Е и Ё (можно использовать любую)
- Ъ, Ь и Ы (можно использовать любую)

Не пиши никаких рассуждений, дай сразу же
→ окончательный ответ в формате
Ответ: <ваш ответ>.

ОБЯЗАТЕЛЬНО используй точный формат
→ окончательного ответа:

Пример угадывания буквы:
Ответ: О

Пример угадывания слова:
Ответ: СЛОВО: СОБАКА

ВАЖНО:
- Окончательный ответ должен начинаться со
→ слова "Ответ:".
- Не добавляй ничего после угаданной буквы
→ или слова.
- Не задавай вопросов и не комментируй мой
→ ход.

****Текущее состояние:****
Слово: {revealed word}
Использовано попыток: {turns used}/{maximum
→ turns}

Начинаем игру! Ваш ход.

Wheel of Fortune – CoT – English

[user]
Let's play "Wheel of Fortune".

In this game, you need to guess a hidden word.
→ You can guess letters one at a time or try to
→ guess the whole word. When you guess a
→ letter, if it is in the word, it will be revealed.
→ If the letter appears multiple times in the
→ word, all occurrences will be revealed. You
→ win either by guessing the whole word or by
→ guessing all its letters. You lose if you run
→ out of attempts without guessing the word.

****Rules:****
- You have {maximum turns} attempts

Explain your reasoning before answering in the
→ format:

Reasoning: <your thoughts>

Answer: <your answer>

You MUST use the exact final answer format!

Example of guessing a letter:

Answer: E

Example of guessing a word:

Answer: WORD: EXAMPLE

IMPORTANT:

- The final answer must start with the word
→ "Answer:".
- Do not add anything after the guessed letter or
→ word.
- Do not ask questions or comment on my move.

****Current state:****

Word: {revealed word}

Attempts used: {turns used}/{maximum turns}

Let's start the game! Your move.

Wheel of Fortune – CoT – Russian

[user]

Давай сыграем в "Поле чудес".

В этой игре от вас требуется угадать скрытое
→ слово. Вы можете угадывать буквы по
→ одной или пытаться угадать слово
→ целиком. Когда вы угадываете букву, если
→ она есть в слове, она открывается. Если
→ буква присутствует в слове несколько раз,
→ откроются все её вхождения. Вы
→ выигрываете либо угадав слово целиком,
→ либо угадав все его буквы. Вы
→ проигрываете, если исчерпаете все свои
→ попытки, не угадав слово.

****Правила:****

- У вас есть {maximum turns} попыток
- ****Эквивалентные буквы****: Следующие буквы
→ считаются взаимозаменяемыми:
 - И и Й (можно использовать любую)
 - Е и Ё (можно использовать любую)
 - Ъ, Ь и Ы (можно использовать любую)

Объясни свои рассуждения перед ответом в
→ формате:

Рассуждение: <ваши мысли>

Ответ: <ваш ответ>

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа:

Пример угадывания буквы:

Ответ: О

Пример угадывания слова:

Ответ: СЛОВО: СОБАКА

ВАЖНО:

- Окончательный ответ должен начинаться со
→ слова "Ответ:".

- Не добавляй ничего после угаданной буквы
→ или слова.

- Не задавай вопросов и не комментируй мой
→ ход.

****Текущее состояние:****

Слово: {revealed word}

Использовано попыток: {turns used}/{maximum
→ turns}

Начинаем игру! Ваш ход.

L.1.6 20 Features

20 Features – AO – English

[user]

You have perfect mastery of words and their
→ feature representations. You are playing the
→ 'Guess the Word' game. An object has been
→ secretly chosen, and on each turn one new
→ feature of this object is revealed to you.
Goal: guess the word based on these features in
→ minimal number of moves. Keep in mind that
→ the number of moves is limited.

Each of your responses must start with the word
→ 'ANSWER', followed by ONLY one guessed
→ word — no questions, explanations, or extra
→ text:

(example: 'ANSWER: apple')

If you say the right word, the game ends. If the
→ game isn't over, then you still haven't said
→ the right word, and you need to try another
→ one. There is no point in repeating the words.
Never ask questions, don't explain, don't write
→ extra text. Only one final answer after
→ 'ANSWER'.

Your turn:

{feature name}: {feature value}

20 Features – AO – Russian

[user]

Ты в совершенстве владеешь словами и их
→ признаковым представлением. Ты играешь
→ в игру 'Угадай слово'. Загадан объект, и
→ каждый ход тебе раскрывается один
→ новый признак этого объекта.

Цель — угадать слово на основе этих признаков
→ за минимальное число ходов. Учитывай,
→ что количество ходов ограничено.

Каждый твой ответ должен начинаться со
→ слова 'ОТВЕТ', после которого следует
→ ТОЛЬКО одно угаданное слово — никаких
→ вопросов, объяснений или дополнительного
→ текста:

(пример: 'ОТВЕТ: яблоко')

Если ты называешь угаданное слово, игра
→ заканчивается. Если игра не закончилась,
→ значит ты до сих пор не называл
→ правильное слово, и нужно попробовать
→ другое. Повторять слова смысла нет.

Никогда не задавай вопросов, не объясняй, не
→ пиши лишнего текста. Только один
→ финальный ответ после 'ОТВЕТ'.
Твой ход:

{feature name}: {feature value}

20 Features – CoT – English

[user]

You have perfect mastery of words and their
→ feature representations. You are playing the
→ 'Guess the Word' game. An object has been
→ secretly chosen, and on each turn one new
→ feature of this object is revealed to you.
After each feature, you must analyze all
→ accumulated information, remember all the
→ words you have already said, logically justify
→ possible hypotheses and after the word
→ 'ANSWER' write one guessed word and
→ write nothing else after that.

Goal: guess the word based on these features in
→ minimal number of moves. Keep in mind that
→ the number of moves is limited.

Before each response, MANDATORY think
→ through and write step-by-step combination
→ of known features, analyze them and make a
→ logical conclusion. Only in the very end
→ specify the guessed word after 'ANSWER'
→ and write nothing else after that:
(example: 'Color — red, shape — round, grows on
→ a tree, used in pies. Fits all features.
→ ANSWER: apple')

If you say the right word, the game ends. If the
→ game isn't over, then you still haven't said
→ the right word, and you need to try another
→ one. There is no point in repeating the words.
Never ask questions, don't write extra text. Only
→ logical justification and one final answer after
→ 'ANSWER'.

Your turn:

{feature name}: {feature value}

20 Features – CoT – Russian

[user]

Ты в совершенстве владеешь словами и их
→ признаковым представлением. Ты играешь
→ в игру 'Угадай слово'. Загадан объект, и
→ каждый ход тебе раскрывается один
→ новый признак этого объекта.

После каждого признака ты должен
→ проанализировать всю накопленную
→ информацию, вспомнить все слова,
→ которые называл, логически обосновать
→ возможные гипотезы и после слова
→ 'ОТВЕТ' написать одно угаданное слово
→ и после этого больше ничего не писать.

Цель — угадать слово на основе этих признаков
→ за минимальное число ходов. Учитывай,
→ что количество ходов ограничено.

Перед каждым ответом ОБЯЗАТЕЛЬНО

→ обдумай и запиши пошагово сочетание уже
→ известных признаков, проанализируй их и
→ сделай логический вывод. Только в самом
→ конце укажи угаданное слово после слова
→ 'ОТВЕТ', и после этого больше ничего не
→ пиши:

(пример: 'Цвет — красный, форма — круглая,
→ растёт на дереве, используется в пирогах.
→ Подходит по всем признакам. ОТВЕТ:
→ яблоко')

Если ты называешь угаданное слово, игра
→ заканчивается. Если игра не закончилось,
→ значит ты до сих пор не называл
→ правильное слово, и нужно попробовать
→ другое. Повторять слова смысла нет.
Никогда не задавай вопросов, не пиши лишнего
→ текста. Только логическое обоснование и
→ один финальный ответ после 'ОТВЕТ'.
Твой ход:

{feature name}: {feature value}

L.1.7 Alias

Alias – AO – English

[user]

You have perfect knowledge about various people.
→ You are playing the 'Alias' game. A famous
→ person has been secretly chosen.

Each turn you can ask ONE question that can be
→ answered ONLY with 'yes' or 'no'. Goal:
→ guess the person in minimal number of
→ questions. Keep in mind that the number of
→ turns is limited.

Each of your responses must start with the word
→ 'ANSWER', followed by either ONE
→ question or the final person, and write
→ nothing else after that:

(example of a question: 'ANSWER: Is this a
→ politician?') - put question mark '?' at the
→ end of your question

(example of final answer: 'ANSWER: Angelina
→ Jolie!') - put exclamation mark '!' at the
→ end of your final answer

Ask questions to get maximum detailed and
→ diverse information in minimal number of
→ turns. Don't explain, don't write extra text.

Your turn:

Alias – AO – Russian

[user]

Ты в совершенстве владеешь информацией о
→ разных людях. Ты играешь в игру
→ 'Алиас'. Загадан известный человек.

Каждый ход ты можешь задавать ОДИН
→ вопрос, на который можно ответить
→ ТОЛЬКО 'да' или 'нет'. Цель — угадать
→ человека за минимальное число вопросов.
→ Учитывай, что количество ходов
→ ограничено.

Каждый твой ответ должен начинаться со
→ слова 'ОТВЕТ', после него укажи либо
→ ОДИН вопрос, либо итогового человека, и
→ после этого больше ничего не пиши:

(пример одного вопроса: 'ОТВЕТ: Это
 ↳ политик?') - в конце вопроса ставь
 ↳ вопросительный знак '?'
 (пример финального ответа: 'ОТВЕТ:
 ↳ Анджелины Джоли!') - в конце финального
 ↳ ответа ставь восклицательный знак '!'
 Задавай вопросы таким образом, чтобы за
 ↳ минимальное их количество получить
 ↳ максимально подробную и
 ↳ разностороннюю информацию. Не
 ↳ объясняй, не пиши лишнего.

Твой ход:

Alias – CoT – English

[user]
 You have perfect knowledge about various people.
 ↳ You are playing the 'Alias' game. A famous
 ↳ person has been secretly chosen.
 Each turn you can ask ONE question that can be
 ↳ answered ONLY with 'yes' or 'no'. Goal:
 ↳ guess the person in minimal number of
 ↳ questions. Keep in mind that the number of
 ↳ turns is limited.
 Before each answer and question, MANDATORY
 ↳ think through and write step-by-step
 ↳ combination of already known questions and
 ↳ answers, analyze them, make assumptions
 ↳ about the possible persons chosen and what
 ↳ information is missing, and only then in the
 ↳ very last sentence after the word
 ↳ 'ANSWER' specify either ONE question or
 ↳ the final person, and write nothing else after
 ↳ that: (example of a question: 'This is a
 ↳ famous and influential person. ANSWER: Is
 ↳ this a politician?') - put question mark '?'
 ↳ at the end of your question
 (example of final answer: 'This is an actress and
 ↳ Hollywood star. ANSWER: Angelina Jolie!')
 ↳ - put exclamation mark '!' at the end of your
 ↳ final answer
 Ask questions to get maximum detailed and
 ↳ diverse information in minimal number of
 ↳ turns.

Your turn:

Alias – CoT – Russian

[user]
 Ты в совершенстве владеешь информацией о
 ↳ разных людях. Ты играешь в игру
 ↳ 'Алиас'. Загадан известный человек.
 Каждый ход ты можешь задавать ОДИН
 ↳ вопрос, на который можно ответить
 ↳ ТОЛЬКО 'да' или 'нет'. Цель — угадать
 ↳ человека за минимальное число вопросов.
 ↳ Учитывай, что количество ходов
 ↳ ограничено.

Перед каждым ответом и вопросом
 ↳ ОБЯЗАТЕЛЬНО обдумай и запиши
 ↳ пошагово сочетание уже известных тебе
 ↳ вопросов и ответов на них, проанализируй,
 ↳ сделай предположения о возможных
 ↳ вариантах и о том, какой информации не
 ↳ хватает, и только затем в самом последнем
 ↳ предложении после слова 'ОТВЕТ' укажи
 ↳ либо ОДИН вопрос, либо итогового
 ↳ человека, и после этого больше ничего не
 ↳ пиши: (пример одного вопроса: 'Это
 ↳ известная и влиятельная личность.
 ↳ ОТВЕТ: Это политик?') - в конце вопроса
 ↳ ставь вопросительный знак '?'
 (пример финального ответа: 'Это актриса и
 ↳ звезда Голливуда. ОТВЕТ: Анджелины
 ↳ Джоли!') - в конце финального ответа
 ↳ ставь восклицательный знак '!'
 Задавай вопросы таким образом, чтобы за
 ↳ минимальное их количество получить
 ↳ максимально подробную и
 ↳ разностороннюю информацию.

Твой ход:

L.2 Judge Prompts

Situation Puzzles – judge (English and Russian)

[English -- yes/no question validation]
 [user]

You are the host of a game of “yes/no riddles”
 ↳ (also known as situation puzzles or lateral
 ↳ thinking puzzles).
 You have full information about the riddle: the
 ↳ setup and the solution.

SETUP:
 {setup}

SOLUTION:
 {solution}

DIALOGUE HISTORY:
 {dialogue_history}

Rules:

- The player asks questions that are answered
 ↳ with “Yes”, “No”, or “It doesn't matter”.
- The final answer must be strictly one of the
 ↳ following options:
 - 1) Yes
 - 2) No
 - 3) It doesn't matter
 - 4) The question cannot be answered
 ↳ unambiguously with yes/no
 - 5) Ask only one question
- Do not reveal the solution directly.
- If the answer to the question is contained in the
 ↳ setup or in the solution, write “Answer: Yes”
 ↳ or “Answer: No”.
- If you do not know the correct answer to the
 ↳ question or there is no information about it,
 ↳ write “Answer: It doesn't matter.”

- If the question cannot be answered with "Answer: Yes" or "Answer: No", write "Answer: The question cannot be answered unambiguously with yes/no." For example, if it is a question that already offers answer options, or if the question contains a hidden assertion (often false) that you would automatically agree with by answering "Yes" or "No".
- If there are multiple questions, write "Answer: Ask only one question"
- Do not nitpick over tiny details. For example, "guard" and "security officer" are the same thing.

Write nothing except the answer to the question. You MUST use the exact final answer format:
 Answer: <Yes, No, It doesn't matter, The question cannot be answered unambiguously with yes/no, Ask only one question>

IMPORTANT:

- The final answer must begin with the word "Answer:".
- Do not add anything after the answer.

Player's question: "{question}"

[English -- final answer validation]
 [user]

You are the judge in a game of "yes/no riddles" (also known as situation puzzles or lateral thinking puzzles). You are given:

- the story setup,
- the true solution,
- the player's guess,
- a key fact from the true solution that must be present in a correct guess.

SETUP:
 {{ setup }}

TRUE SOLUTION:
 {{ solution }}

PLAYER'S GUESS:
 {{ guess }}

KEY FACT:
 {{ facts }}

DIALOGUE HISTORY:
 {{ dialogue_history }}

Evaluation rules:

1. The riddle is considered solved if the player's guess contains the key part of the true solution. The key part of the true solution is localized in the key fact.
2. Consider the riddle solved (True) if the guess explicitly contains this fact or if the same information follows from it.
3. Different wordings are acceptable if the meaning matches.
4. Extra details do not hurt and are not as important, as long as they do not contradict the true solution or have roughly the same meaning.

For example, in this riddle:

Setup: "There was a long line at the entrance to a recording studio for tickets to the taping of a show. A man without a ticket approached the entrance, and people not only did not object, but let him go ahead. Why?"

True solution: "The man was not an audience member. He was a studio employee and was carrying a reel of magnetic tape for the recording. Security let him through because he was bringing important work material for the show."

Key fact: "The man was a studio employee and was carrying a reel of magnetic tape."

Here, all guesses that in some way contain the information that the man was an employee can be considered a valid solution.

- Write nothing except the answer to the question.
- Do not add anything after the answer.

You MUST use the exact final answer format:
 Answer: <True, False>

[Russian -- yes/no question validation]
 [user]

Ты — ведущий игры в данетки.
 У тебя есть полная информация о загадке:
 → завязка и разгадка.

ЗАВЯЗКА:
 {setup}

РАЗГАДКА:
 {solution}

ИСТОРИЯ ДИАЛОГА:
 {dialogue_history}

Правила:

- Игрок задает вопросы, на которые отвечают "Да", "Нет" или "Неважно".
- Финальный ответ должен быть строго одним из вариантов:
 - 1) Да
 - 2) Нет
 - 3) Неважно
 - 4) На вопрос нельзя ответить однозначно → да/нет. Переформулируй его.
 - 5) Задай только один вопрос
- Не раскрывай разгадку напрямую.
- Если ответ на вопрос содержится в завязке или в разгадке, то пиши "Ответ: Да" или "Ответ: Нет".
- Если ты не знаешь правильный ответ на вопрос или нет информации про это, то пиши "Ответ: Неважно"
- Если на вопрос нельзя ответить "Да" или "Нет", то пиши "Ответ: На вопрос нельзя ответить однозначно да/нет. Переформулируй его.". Например, если это вопросы в которых уже предложены варианты ответа или если вопрос содержит в себе скрытое утверждение (часто ложное), с которым вы автоматически соглашаетесь, отвечая "Да" или "Нет".
- Если вопросов несколько, то пиши "Ответ: Задай только один вопрос"

- Не надо докапываться до мельчайших
 - деталей. Например, "охранник" и
 - "сотрудник службы безопасности" - это
 - одно и то же.

Ничего не пиши кроме ответа на вопрос.

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа:

Ответ: <Да, Нет, Неважно, На вопрос нельзя
→ ответить однозначно да/нет, Задай только
→ один вопрос>

ВАЖНО:

- Окончательный ответ должен начинаться со
 - слова "Ответ:".
- Не добавляй ничего после ответа.

Вопрос игрока: "{question}"

[Russian -- final answer validation]
[user]

Ты — судья в игре «данетки». Тебе даны:

- завязка истории,
- истинная разгадка,
- догадка игрока,
- ключевой факт из истинной разгадки,
 - который обязательно должен
 - присутствовать в корректной догадке.

ЗАВЯЗКА:

{{ setup }}

ИСТИННАЯ РАЗГАДКА:

{{ solution }}

ДОГАДКА ИГРОКА:

{{ guess }}

КЛЮЧЕВОЙ ФАКТ:

{{ facts }}

ИСТОРИЯ ДИАЛОГА:

{{ dialogue_history }}

Правила оценки:

1. Данетка считается разгаданной, если в
 - догадке игрока содержится ключевая
 - часть истинной разгадки. Ключевая часть
 - истинной разгадки локализована в
 - ключевом факте.
2. Считай, что данетка разгадана (True), если
 - догадка явно содержит этот факт или из
 - нее следует та же информация.
3. Разные формулировки допустимы, если
 - смысл совпадает.
4. Лишние детали не мешают и не так важны,
 - если они не противоречат истинной
 - разгадке или имеют примерно один и тот
 - же смысл.

Например, в такой данетке:

Завязка: "У входа в студию звукозаписи стояла
→ длинная очередь за билетами на запись
→ передачи. Мужчина без билета подошёл ко
→ входу, и люди не только не возмутились, а
→ сами пропустили его вперёд. Почему?"

Истинная разгадка: "Мужчина не был

- зрителем. Он был сотрудником студии и
- нёс катушку с магнитной лентой для
- записи. Охрана пропустила его потому что
- он вёз важный рабочий материал для
- передачи."

Ключевой факт: "Мужчина был сотрудником

- студии и нёс катушку с магнитной
- лентой."

Здесь за валидную разгадку можно считать все

- догадки, которые как-то в себе содержат
- информацию о том, что мужчина был
- сотрудником.

- Ничего не пиши кроме ответа на вопрос.

- Не добавляй ничего после ответа.

ОБЯЗАТЕЛЬНО используй точный формат

→ окончательного ответа:

Ответ: <True, False>

20 Features – judge (English and Russian)

[English -- answer validation]
[user]

You are a strict but fair judge in the "Guess the
→ Word" game.

Your task is to compare the target word with the
→ player's answer and determine if it is
→ essentially correct.

Consider the answer CORRECT ("true") if:

- The words match exactly.
- It is the same word in different grammatical
 - forms (plural/singular, case, gender).
- They are generally accepted exact synonyms or
 - words that essentially mean the same thing
 - (e.g., hippo and hippopotamus).
- The difference is only in spelling (e.g.,
 - color/colour, a minor typo).
- It is a collective material and a single instance of
 - that same object (e.g., pearls and pearl, peas
 - and pea).
- It is a diminutive or affectionate form of the
 - exact same word (e.g., drop and droplet, dog
 - and doggy).

Consider the answer INCORRECT ("false") if:

- They are completely different objects.
- The player's answer is a broader category
 - (target: "apple", answer: "fruit" or "food").
- The player's answer is only a part of the object
 - (target: "tree", answer: "leaf" or "branch").
- It is merely an association (target: "rain",
 - answer: "umbrella").
- An animal and its cubs are different objects.

Reply ONLY with "true" or "false". No

→ explanations, periods, quotes, or reasoning.

Examples:

Target word: apple | Player's answer: apples ->
→ true

Target word: hippo | Player's answer:
→ hippopotamus -> true

Target word: hedgehog | Player's answer:
→ hedgehogs -> true

Target word: automobile | Player's answer: car ->
→ true
Target word: pearl | Player's answer: pearls ->
→ true
Target word: apple | Player's answer: pear ->
→ false
Target word: apple | Player's answer: fruit ->
→ false
Target word: tree | Player's answer: branch ->
→ false
Target word: doctor | Player's answer: hospital ->
→ false
Target word: baby elephant | Player's answer:
→ elephant -> false
Target word: drop | Player's answer: droplet ->
→ true

Target word: {target_word} | Player's answer:
→ {player_answer} ->

[Russian -- answer validation]
[user]

Ты — строгий, но справедливый судья в игре
→ "Угадай слово".
Твоя задача — сравнить загаданное слово с
→ ответом игрока и определить, верный ли
→ он по сути.

Считай ответ **ВЕРНЫМ** ("true"), если:
- Слова полностью совпадают.
- Это одно и то же слово в разных формах
→ (падеж, число, род).
- Это общепринятые точные синонимы или
→ слова, которые по сути означают одно и то
→ же (например, бегемот и гиппопотам).
- Разница только в орфографии (например, е/ё,
→ мелкая опечатка).
- Это собирательное понятие и единичный
→ экземпляр того же объекта (например,
→ жемчуг и жемчужина, горох и горошина).
- Это то же самое слово в
→ уменьшительно-ласкательной форме
→ (например, дом и домик, мост и мостик).

Считай ответ **НЕВЕРНЫМ** ("false"), если:
- Это совершенно разные объекты.
- Ответ игрока — это более широкая категория
→ (загадано "яблоко", ответ "фрукт" или
→ "еда").
- Ответ игрока — это лишь часть объекта
→ (загадано "дерево", ответ "лист").
- Это просто ассоциация (загадано "дождь",
→ ответ "зонт").
- Животное и его детёныши - это разные
→ объекты

В ответ напиши **ТОЛЬКО** "true" или "false".
→ Никаких объяснений, точек, кавычек или
→ рассуждений.

Примеры:

Загаданное слово: яблоко | Ответ игрока:
→ яблоки -> true
Загаданное слово: бегемот | Ответ игрока:
→ гиппопотам -> true
Загаданное слово: ёж | Ответ игрока: ежа ->
→ true

Загаданное слово: автомобиль | Ответ игрока:
→ машина -> true
Загаданное слово: жемчужина | Ответ игрока:
→ жемчуг -> true
Загаданное слово: яблоко | Ответ игрока:
→ груша -> false
Загаданное слово: яблоко | Ответ игрока:
→ фрукт -> false
Загаданное слово: дерево | Ответ игрока: ветка
→ -> false
Загаданное слово: врач | Ответ игрока:
→ больница -> false
Загаданное слово: слонёнок | Ответ игрока:
→ слон -> false
Загаданное слово: мост | Ответ игрока: мостик
→ -> true

Загаданное слово: {target_word} | Ответ
→ игрока: {player_answer} ->

Alias – judge (English and Russian)

[English -- identity guess validation]
[user]

You are a strict, intelligent, and fair judge in a
→ "Guess the Persona" game. Your task is to
→ analyze the player's message and determine
→ whether they successfully guessed the secret
→ target persona.

RULES AND LOGIC:

1. If the player's message is a clarifying question
→ about the persona's traits (e.g., "Is it a
→ man?", "Did he star in movies?", "Is this
→ person still alive?"), you must answer
→ "Error".
2. If the player's message is an attempt to guess a
→ specific person (e.g., "Is it Brad Pitt?", "My
→ guess is Napoleon", "George Washington"),
→ you must compare this guessed person with
→ the Target Persona.
3. Account for ANY name variations of the exact
→ same person. Consider it a match ("Yes") if
→ the player uses:
 - Full, short, or initialized forms (e.g., JFK =
→ John F. Kennedy = John Kennedy).
 - Real names and stage names/pseudonyms
→ (e.g., Marilyn Monroe = Norma Jeane
→ Mortenson; Mark Twain = Samuel
→ Clemens).
 - Different historical names, aliases, or titles of
→ the same person (e.g., Octavian =
→ Augustus Caesar; Pope Francis = Jorge
→ Mario Bergoglio).
 - Maiden/married names, provided they point
→ to the exact same person (e.g., Jacqueline
→ Kennedy = Jackie Onassis).
 - Different transliterations from other languages
→ (e.g., Fyodor Dostoevsky = F. Dostoevski).
 - Different languages (e.g., Elon Musk = Илон
→ Маск)
4. If the player names a **DIFFERENT** person
→ (even a relative or someone with the same
→ last name), answer "No".
5. If the player's answer is not an attempt to
→ guess a specific person, answer "Error".

OUTPUT FORMAT:

Your response MUST consist STRICTLY OF
→ ONE WORD: "Yes", "No" or "Error". Any
→ comments, explanations, periods, or other
→ punctuation marks are strictly forbidden.

EXAMPLES:

Target Persona: Angelina Jolie
Player's Message: Is she an actress?
Your Response: Error

Target Persona: Angelina Jolie
Player's Message: Is it Angelina Jolie?
Your Response: Yes

Target Persona: John F. Kennedy
Player's Message: I think it's JFK.
Your Response: Yes

Target Persona: Muhammad Ali
Player's Message: Is it Cassius Clay?
Your Response: Yes

Target Persona: Mark Twain
Player's Message: Samuel Clemens
Your Response: Yes

Target Persona: Elon Musk
Player's Message: Is this person related to space?
Your Response: Error

Target Persona: Snoop Dogg
Player's Message: Calvin Cordozar Broadus Jr.
Your Response: Yes

Target Persona: Snoop Dogg
Player's Message: Снуп Догг
Your Response: Yes

Target Persona: Theodore Roosevelt
Player's Message: Franklin D. Roosevelt
Your Response: No

Target Persona: Queen Elizabeth II
Player's Message: Queen Victoria
Your Response: No

Target Persona: Julius Caesar
Player's Message: Maybe it's Caesar?
Your Response: Yes

Target Persona: Michael Jackson
Player's Message: Is it Michael Jordan?
Your Response: No

ANALYZE THE PLAYER'S MOVE:

Target Persona: {target_person}
Player's Message: {model_answer}
Your Response:

[English -- biography question answering]
[user]

Role:

You are a strict and accurate judge in the "Guess
→ the Person" game. Your sole purpose is to
→ answer the player's closed-ended questions
→ about the secret person.

Inputs:

1. The name of the secret person.
2. A biographical summary of this person.
3. The player's question.

Answering Rules (CRITICALLY → IMPORTANT):

1. Your answer must consist of exactly one word:
→ **Yes** or **No**. Any explanations,
→ comments, or punctuation marks are strictly
→ forbidden.
2. Rely exclusively on the provided biographical
→ summary, the Target name, and basic logic.
3. If the summary does not contain the
→ information needed to answer a factual
→ question, answer **No**.
4. If the question requires a detailed answer (e.g.,
→ "Where was he born?") or cannot be
→ answered with a simple "yes/no", answer
→ **No**.

Special Instructions for Evaluating Facts:

- * **Direct Name Guessing:** Sometimes the
→ player may try to guess the person directly
→ (e.g., with a question like "Is it Adam
→ Smith?" or a statement like "Adam Smith!").
→ In this case, compare the player's guess with
→ the "Target Name". If the player named the
→ correct person (typos, pseudonyms, or using
→ only the last name if it unambiguously points
→ to the Target Name are allowed), answer
→ "Yes". If a completely different person is
→ named, answer "No".
- * **Primary Identity vs. Secondary Roles:** If
→ the player asks a general question about a
→ profession ("Is he a politician?", "Is he a
→ writer?"), answer "Yes" ONLY if this is the
→ **main** reason for their fame or is
→ inextricably linked to it (for example, Bob
→ Woodward is a journalist by profession, but
→ he also writes books, and has written many
→ famous ones — he is a writer, answer "Yes").
→ HOWEVER, if a person became famous in
→ one field, and another was temporary or
→ secondary (for example, actor Arnold
→ Schwarzenegger was a governor), answer
→ "No" to the direct question "Is he a
→ politician?" to avoid misleading the player.
→ Answer "Yes" only to specific questions
→ ("Did he ever hold political office?"). Hobbies
→ and facts from youth (a famous chef played
→ football in his youth) also get a "No" to
→ broad questions ("Is he an athlete?").
→ However, if the question is specific ("Did he
→ play football in his youth?"), then answer
→ "Yes".
- * **Chronology and Centuries:** Pay extremely
→ close attention to the dates of life. Players
→ often ask about centuries. Before answering,
→ mentally match the years of life with the
→ corresponding centuries:
 - * 1801–1900 is the 19th century.
 - * 1901–2000 is the 20th century.
 - * 2001–2100 is the 21st century.
- * If a person lived across the turn of a century
→ (e.g., born in 1895, died in 1960), the
→ correct answer to both "Did this person
→ live in the 19th century?" and "Did this
→ person live in the 20th century?" is "Yes".

****Examples:****

[Example 1]

Name: Arnold Schwarzenegger

Summary: American bodybuilder, actor, and
→ producer. 38th Governor of California.

Question: Is this a politician?

Answer: No

[Example 2]

Name: Arnold Schwarzenegger

Summary: American bodybuilder, actor, and
→ producer. 38th Governor of California.

Question: Did he ever hold political office?

Answer: Yes

[Example 3]

Name: Bob Woodward

Summary: American investigative journalist.

→ Author of numerous non-fiction books about
→ US presidents.

Question: Is he a writer?

Answer: Yes

[Example 4]

Name: Gordon Ramsay

Summary: A famous British chef, restaurateur,
→ and television presenter. In his youth, he was
→ seriously involved in football and even played
→ for the Rangers youth team, but due to an
→ injury, he was forced to leave sports and take
→ up cooking.

Question: Is this an athlete?

Answer: No

[Example 5]

Name: Gordon Ramsay

Summary: A famous British chef, restaurateur,
→ and television presenter. In his youth, he was
→ seriously involved in football and even played
→ for the Rangers youth team, but due to an
→ injury, he was forced to leave sports and take
→ up cooking.

Question: Did this person try to build a sports
→ career before becoming famous?

Answer: Yes

[Example 6]

Name: Vladimir Lenin

Summary: Russian revolutionary, major Marxist
→ theoretician, Soviet political and state figure.
→ Years of life: 1870 — 1924.

Question: Did this person live in the 18th century?

Answer: No

[Example 7]

Name: Vladimir Lenin

Summary: Russian revolutionary, major Marxist
→ theoretician, Soviet political and state figure.
→ Years of life: 1870 — 1924.

Question: Did this person live in the 19th century?

Answer: Yes

[Example 8]

Name: Marilyn Monroe

Summary: American actress, 1950s sex symbol,
→ singer, and model.

Question: What year was she born?

Answer: No

[Example 9]

Name: Adam Smith

Summary: Scottish economist and moral
→ philosopher, a pioneer of political economy.

Question: Is this Adam Smith?

Answer: Yes

[Example 10]

Name: Adam Smith

Summary: Scottish economist and moral
→ philosopher, a pioneer of political economy.

Question: Karl Marx!

Answer: No

ANSWER THE PLAYER'S QUESTION

Target Name: {target_name}

Player's Question: {question}

START OF BIOGRAPHY

{bio_text}

END OF BIOGRAPHY

Target Name: {target_name}

Player's Question: {question}

Your Answer:

[Russian -- identity guess validation]

[user]

Ты — строгий, но умный и справедливый судья

→ в игре "Угадай личность". Твоя задача —

→ проанализировать реплику игрока и

→ определить, смог ли он назвать

→ загаданную личность.

ПРАВИЛА И ЛОГИКА:

1. Если реплика игрока — это уточняющий

→ вопрос о свойствах персонажа (например,

→ "Это мужчина?", "Он снимался в кино?",

→ "Этот человек жив?"), ты должен

→ ответить «Ошибка».

2. Если реплика игрока — это попытка угадать

→ конкретную персону (например, "Это Брэд

→ Питт?", "Мой ответ: Наполеон", "Джордж

→ Вашингтон"), ты должен сравнить эту

→ персону с Загаданной личностью.

3. Учитывай любые вариации имени одного и

→ того же человека. Считай совпадением

→ («Да»):

- Полные и краткие формы (А. С. Пушкин =

→ Александр Сергеевич Пушкин =

→ Пушкин).

- Настоящие имена и псевдонимы (Мэрилин

→ Монро = Норма Джин Бейкер; Марк

→ Твен = Сэмюэл Клеменс).

- Разные исторические имена и титулы

→ одного человека (Бенедикт Спиноза =

→ Барух Спиноза; Владимир I = Владимир

→ Святославович = Владимир Красное

→ Солнышко; Петр I = Петр Великий).

- Разные языки и транслитерации (Elon Musk

→ = Илон Маск).

- Замужние/девичьи фамилии, если они

→ однозначно указывают на этого

→ человека.

4. Если игрок назвал ДРУГОГО человека

→ (даже однофамильца или родственника),

→ отвечай «Нет».

5. Если ответ игрока не является попыткой
→ угадать конкретную персону, отвечай
→ «Ошибка».

ФОРМАТ ОТВЕТА:

Твой ответ должен состоять СТРОГО ИЗ
→ ОДНОГО СЛОВА: «Да», «Нет» или
→ «Ошибка». Запрещены любые
→ комментарии, пояснения, точки или другие
→ знаки препинания.

ПРИМЕРЫ:

Загаданная личность: Анджелина Джоли
Ответ игрока: Она актриса?
Твой ответ: Ошибка

Загаданная личность: Анджелина Джоли
Ответ игрока: Это Анджелина Джоли?
Твой ответ: Да

Загаданная личность: А. С. Пушкин
Ответ игрока: Я думаю, это Александр
→ Сергеевич Пушкин.
Твой ответ: Да

Загаданная личность: Иосиф Сталин
Ответ игрока: Это Иосиф Джугашвили?
Твой ответ: Да

Загаданная личность: Бенедикт Спиноза
Ответ игрока: Барух Спиноза
Твой ответ: Да

Загаданная личность: Илон Маск
Ответ игрока: Этот человек связан с космосом?
Твой ответ: Ошибка

Загаданная личность: Snoor Dogg
Ответ игрока: Снуп Догг
Твой ответ: Да

Загаданная личность: Лев Толстой
Ответ игрока: Алексей Толстой
Твой ответ: Нет

Загаданная личность: Елизавета II
Ответ игрока: Королева Виктория
Твой ответ: Нет

Загаданная личность: Гай Юлий Цезарь
Ответ игрока: Возможно, это Цезарь?
Твой ответ: Да

Загаданная личность: Папа Римский Франциск
Ответ игрока: Хорхе Марио Бергольо
Твой ответ: Да

Загаданная личность: Майкл Джексон
Ответ игрока: Это Майкл Джордан?
Твой ответ: Нет

ПРОАНАЛИЗИРУЙ ХОД ИГРОКА:
Загаданная личность: {target_person}
Ответ игрока: {model_answer}
Твой ответ:

[Russian -- biography question answering]
[user]

Роль:

Ты — строгий и точный судья в игре "Угадай
→ персону". Твоя единственная цель —
→ отвечать на закрытые вопросы игрока о
→ загаданном человеке.

****Входные данные:****

1. Имя загаданной персоны.
2. Справка с информацией об этой персоне.
3. Вопрос игрока.

****Правила ответа (КРИТИЧЕСКИ
→ ВАЖНО):****

1. Твой ответ должен состоять ровно из одного
→ слова: ****Да**** или ****Нет****. Любые
→ пояснения, комментарии или знаки
→ препинания строго запрещены.
2. Опирайся исключительно на
→ предоставленную справку, Загаданное имя
→ и базовую логику.
3. Если в справке нет информации для ответа
→ на вопрос о фактах — отвечай ****Нет****.
4. Если вопрос подразумевает развернутый
→ ответ (например, "Где он родился?") или
→ не может быть отвечен "да/нет" — отвечай
→ ****Нет****.

****Специальные инструкции по оценке
→ фактов:****

- * ****Прямое угадывание имени:**** Иногда
→ игрок может попытаться сразу назвать
→ персону (например, вопросом "Это Адам
→ Смит?" или утверждением "Адам Смит!").
→ В этом случае сравни догадку игрока с
→ полем "Загаданное имя". Если игрок
→ назвал правильного человека (допускаются
→ опечатки, использование псевдонимов или
→ указание только фамилии, если она
→ однозначно указывает на Загаданное имя)
→ — отвечай "Да". Если назван совершенно
→ другой человек — отвечай "Нет".
- * ****Главная идентичность против вторичных
→ ролей:**** Если игрок задаёт общий вопрос
→ о профессии ("Он политик?", "Он
→ писатель?"), отвечай "Да" ТОЛЬКО если
→ это ****главная**** причина его известности
→ или неразрывно с ней связана (например,
→ Боб Вудворд по профессии журналист, но
→ при этом пишет книги, написал их много,
→ притом известных — он писатель, ответ
→ "Да"). НО если человек прославился в
→ одной сфере, а другая была временной или
→ вторичной (например, актёр Арнольд
→ Шварценеггер был губернатором), на
→ прямой вопрос "Он политик?" отвечай
→ "Нет", чтобы не увести игрока по ложному
→ следу. Отвечай "Да" только на точные
→ вопросы ("Занимал ли он когда-то
→ политические посты?"). Хобби и факты из
→ юности (знаменитый повар играл в футбол
→ в юности) на общие вопросы ("Он
→ спортсмен?") тоже получают ответ "Нет".
→ Однако, если вопрос звучит детально
→ ("Увлекался ли он футболом в юности?"),
→ то отвечай "Да".
- * ****Хронология и века:**** Будь предельно
→ внимателен к датам жизни. Игроки часто
→ спрашивают про века. Перед ответом
→ мысленно сопоставь годы жизни с веками:
→ * 1801–1900 годы — это XIX (19-й) век.

- * 1901–2000 годы — это XX (20-й) век.
- * 2001–2100 годы — это XXI (21-й) век.
- * Если человек жил на стыке веков
 - (например, родился в 1895, умер в 1960),
 - то на вопросы "Жил ли он в 19 веке?" и
 - "Жил ли он в 20 веке?" правильный
 - ответ — "Да".

****Примеры:****

[Пример 1]

Имя: Арнольд Шварценеггер

Справка: Американский культурист, актёр и
→ продюсер. 38-й губернатор Калифорнии.

Вопрос: Это политик?

Ответ: Нет

[Пример 2]

Имя: Арнольд Шварценеггер

Справка: Американский культурист, актёр и
→ продюсер. 38-й губернатор Калифорнии.

Вопрос: Занимал ли он когда-нибудь

→ политические должности?

Ответ: Да

[Пример 3]

Имя: Боб Вудворд

Справка: Американский

→ журналист-расследователь. Автор
→ множества документальных книг о
→ президентах США.

Вопрос: Он писатель?

Ответ: Да

[Пример 4]

Имя: Гордон Рамзи

Справка: Знаменитый британский шеф-повар,

→ ресторатор и телеведущий. В юности
→ серьезно увлекался футболом и даже
→ играл за молодежный состав клуба
→ «Рейнджерс», но из-за травмы был
→ вынужден оставить спорт и заняться
→ кулинарией.

Вопрос: Это спортсмен?

Ответ: Нет

[Пример 5]

Имя: Гордон Рамзи

Справка: Знаменитый британский шеф-повар,

→ ресторатор и телеведущий. В юности
→ серьезно увлекался футболом и даже
→ играл за молодежный состав клуба
→ «Рейнджерс», но из-за травмы был
→ вынужден оставить спорт и заняться
→ кулинарией.

Вопрос: Пытался ли этот человек построить

→ спортивную карьеру до того, как стал
→ известен?

Ответ: Да

[Пример 6]

Имя: Владимир Ленин

Справка: Российский революционер, крупный

→ теоретик марксизма, советский
→ политический и государственный деятель.
→ Годы жизни: 1870 — 1924.

Вопрос: Этот человек жил в 18 веке?

Ответ: Нет

[Пример 7]

Имя: Владимир Ленин

Справка: Российский революционер, крупный

→ теоретик марксизма, советский
→ политический и государственный деятель.
→ Годы жизни: 1870 — 1924.

Вопрос: Этот человек жил в 19 веке?

Ответ: Да

[Пример 8]

Имя: Мэрилин Монро

Справка: Американская киноактриса,

→ секс-символ 1950-х годов, певица и модель.
Вопрос: В каком году она родилась?

Ответ: Нет

[Пример 9]

Имя: Адам Смит

Справка: Шотландский экономист и

→ философ-этик, один из основоположников
→ современной экономической теории.
Вопрос: Это Адам Смит?

Ответ: Да

[Пример 10]

Имя: Адам Смит

Справка: Шотландский экономист и

→ философ-этик, один из основоположников
→ современной экономической теории.

Вопрос: Карл Маркс!

Ответ: Нет

ОТВЕТЬ НА ВОПРОС ИГРОКА

Загаданное имя: {target_name}

Вопрос игрока: {question}

НАЧАЛО БИОГРАФИИ

{bio_text}

КОНЕЦ БИОГРАФИИ

Загаданное имя: {target_name}

Вопрос игрока: {question}

Твой ответ: