Orthographic Constraint Satisfaction and Human Difficulty Alignment in Large Language Models

Bryan E. Tuck, Rakesh M. Verma
University of Houston
Houston, Texas
betuck@uh.edu, rmverma2@uh.edu

Abstract

Large language models must satisfy hard orthographic constraints during controlled text generation, yet systematic cross-architecture evaluation remains limited. We evaluate 28 configurations spanning three model families (Qwen3, Claude Haiku-4.5, GPT-5-mini) on 58 word puzzles requiring character-level constraint satisfaction. Architectural differences produce substantially larger performance gaps (2.0–2.2×, F1=0.761 vs. 0.343) than parameter scaling within families (83% gain from eightfold scaling), suggesting that constraint satisfaction may require specialized architectural features or training objectives beyond standard language model scaling. Thinking budget sensitivity proves heterogeneous: high-capacity models show strong returns (+0.102 to +0.136 F1), while mid-sized variants saturate or degrade. These patterns are inconsistent with uniform compute benefits. Using difficulty ratings from 10,000 human solvers per puzzle, we establish modest but consistent calibration (r=0.24–0.38) across all families, yet identify systematic failures on common words with unusual orthography ("data", "poop", "loll": 86–95% human success, 89–96% model miss rate). These failures reveal over-reliance on distributional plausibility that penalizes orthographically atypical but constraint-valid patterns, suggesting architectural innovations may be required beyond simply scaling parameters or computational budgets.

Keywords: constrained generation, language models, orthographic constraints, reasoning budgets, human difficulty alignment

1. Introduction

How do large language models satisfy hard orthographic constraints during text generation? Consider a model asked to generate English words using only the letters {A, G, I, L, N, O, W}, where every word must contain W and have at least four letters. This task requires satisfying discrete character-level rules, a fundamental challenge distinct from the semantic pattern matching that dominates language model training (Pulvermüller, 2010). Unlike classification tasks with predefined outputs, constrained generation demands navigation of large combinatorial spaces while maintaining strict constraint adherence (Garbacea and Mei, 2025; Min et al., 2023; Yin and Neubig, 2022). Characterizing how neural networks handle symbolic constraints informs applications requiring structured output (educational word games, controlled content generation, formal language tasks) while probing whether current architectures represent linguistic structure beyond distributional patterns.

Despite extensive research on language model generation capabilities, few studies have systematically examined how models handle hard orthographic constraints in open-vocabulary settings. Prior work on constrained text generation has largely focused on semantic or lexical constraints (Lu et al., 2022), while recent benchmarks suggest models still struggle with explicit character-level rules (Edman et al., 2024). Little prior work has systematically evaluated how model scale, computational budgets, and reasoning modes interact to satisfy character-level constraints with human-calibrated difficulty data.

Constraint satisfaction poses fundamental challenges for neural language models. Models trained on distributional patterns must navigate combinatorial search spaces while verifying discrete symbolic rules (Dziri et al., 2023), a qualitatively different task from predicting contextually appropriate continuations (Andreas, 2022). Distributional plausibility and structural validity can conflict when constraint-valid solutions exhibit atypical orthographic patterns (Liu et al., 2023): a word may be frequent in training data yet appear implausible under specific letter restrictions, or satisfy all constraints yet trigger low generation probabilities due to unusual letter combinations.

We address this gap using word puzzles from the New York Times Spelling Bee, where models generate English words using only specified letter sets with mandatory letter inclusions and minimum length requirements. This task provides three methodological advantages: (1) it isolates constraint-handling from semantic reasoning, enabling focused study of orthographic knowledge; (2) difficulty is calibrated for human solvers, providing ecological validity for model-human comparison; and (3) solver data from 10,000 users per puzzle enables direct calibration without proxy measures.

We conduct an evaluation across three model families spanning 28 configurations from Qwen3, Claude Haiku-4.5, and GPT-5-mini, testing direct generation and reasoning-guided modes with varying computational budgets. Across 58 puzzles, we evaluate 1,624 experiments in zero-shot settings to isolate intrinsic constraint-handling capabilities.

Our contributions are as follows:

1. **Cross-family performance characterization** (Section 5.1): Architectural differences produce substantially larger performance gaps (2.0–2.2$\times$) than parameter scaling within families (83% gain from eightfold increase), with the gap manifesting primarily through recall rather than precision. Constraint satisfaction may require specialized architectural features, training data characteristics, or optimization objectives beyond general language modeling.

2. **Heterogeneous budget sensitivity** (Section 5.2): Thinking budget effects vary dramatically across models: high-capacity variants show strong returns (+0.102 to +0.136 F1), mid-size models degrade monotonically with increased allocation, and the mixture-of-experts 30B variant requires substantial budget for productive expert composition. These patterns are inconsistent with uniform compute scaling.

3. **Human difficulty alignment with mechanistic failure analysis** (Sections 5.3, 5.4): Using 10,000 solver ratings per puzzle, we establish modest but consistent calibration (r=0.24–0.38) across families, yet identify systematic failures on common words with unusual orthography. These failures reveal over-reliance on distributional plausibility that penalizes orthographically atypical but constraint-valid patterns, independent of vocabulary knowledge.

2. Related Work

Orthographic Knowledge in Language Models. Language models encode orthographic structure beyond token frequencies (Itzhak et al., 2022), yet limitations persist: multilingual LLMs over-weight orthographic similarity when processing interlingual homographs (Tanwar et al., 2025), show systematic gaps in grapheme-to-phoneme mapping (Suvarna et al., 2024), and exhibit brittleness to character-level perturbations from subword tokenization (Chai et al., 2024). While character-level (Bunzeck et al., 2024) and byte-level approaches (Dang et al., 2025) avoid tokenization artifacts, they remain less common than subword-based architectures. Collectively, these findings reveal that orthographic knowledge in LLMs remains fragmented and inconsistent, motivating systematic evaluation under explicit constraint satisfaction rather than recognition tasks alone.

Constrained Generation Methods. Existing approaches primarily employ decoding-time guarantees through Grid Beam Search (Hokamp and Liu, 2017), DOMINO (Beurer-Kellner et al., 2024), and grammar-constrained generation (Raspanti et al., 2025; Park et al., 2025), which guarantee constraint satisfaction by restricting the token space during decoding. We evaluate whether instruction-tuned models satisfy constraints through prompt-based guidance alone.

Evaluation with Word Puzzles and Human Difficulty. Word puzzle tasks provide structured testbeds for constraint satisfaction (Giadikiaroglou et al., 2024). Recent work on crossword solving (Saha et al., 2025) demonstrates LLM capabilities on character-constrained puzzles requiring length adherence and character overlaps. Work on lexical complexity prediction (Kelious et al., 2024; Nohejl et al., 2025) shows models can assess word difficulty, but evaluations typically lack human difficulty baselines for generative tasks. Unlike prior work using proxy metrics or small-scale human annotations, we incorporate solver data from 10,000+ users per puzzle to enable direct model-human calibration on constraint satisfaction performance.

3. Experimental Setup

3.1. Task Definition

Models generate English words satisfying explicit orthographic constraints. Each puzzle specifies seven unique letters, one designated as mandatory. Valid outputs must satisfy three constraints: minimum four letters, exclusive use of the seven available letters (repetition permitted), and mandatory inclusion of the designated center letter. For example, given {A, G, I, L, N, O, W} with W as mandatory, “wagon” is valid (uses only available letters, includes W, length ≥ 4) while “along” is invalid (missing W) and “awning” is valid (uses A, W, N, I, N, G from available set, with N repeated).

Words using all seven letters are designated “pangrams.” Puzzles are drawn from a professionally curated collection to ensure vocabulary diversity and appropriate difficulty for human solvers.

3.2. Dataset

We evaluate models on 58 consecutive daily puzzles from the New York Times Spelling Bee (June 2 - July 29, 2025), containing 2,710 total word instances spanning 2,007 unique words across curated, human-verified solution sets. Table 1 presents comprehensive dataset statistics, includ-

ing dataset scale, solution set sizes per puzzle, word length distributions, and pangram counts.

Property	Value/Mean	Range
Words per puzzle	46.7 ± 12.1	22–72
Pangrams per puzzle	1.62 ± 0.83	1–4
Word length (letters)	5.52 ± 1.59	4–13
4-letter words	940	(34.7%)
5-letter words	615	(22.7%)
6+ letter words	1,155	(42.6%)

Table 1: Puzzle and vocabulary characteristics. Solution sets average 47 words with substantial variability (22–72), and longer words (6+ letters) comprise 43% of targets.

3.3. Human Difficulty Data

We augment each puzzle with aggregate performance statistics from the NYT Spelling Bee platform, where over 10,000 users attempt each daily puzzle. For each word, we obtain user success rates (range: 3–97%), providing ground-truth human difficulty estimates calibrated to actual solver performance rather than proxy measures.

3.4. Models and Configurations

We evaluate three model families to examine constrained generation across scales, architectures, and reasoning mechanisms. The selection enables controlled comparison: Qwen3 provides open-source models spanning parameter scales with both dense and mixture-of-experts architectures, while Claude and GPT-5 represent proprietary systems with different reasoning implementations, allowing us to isolate architecture effects from scale effects and open-source from proprietary training regimes.

Qwen3 Family. Five open-source Qwen3 models (Yang et al., 2025) span 4B to 32B parameters: four dense transformers (4B, 8B, 14B, 32B) and one Mixture-of-Experts model (30B-A3B with 3B active parameters). This family enables within-architecture scaling analysis while the MoE variant tests whether sparse expert routing affects constraint satisfaction. We test each model in direct generation mode and thinking mode at three token budgets (4K, 8K, 16K), yielding 20 configurations (5 models × 4: 1 direct + 3 thinking budgets). Thinking mode generates reasoning traces before outputs; direct mode produces responses immediately.

Claude Haiku 4.5. We evaluate Claude Haiku 4.5 (Anthropic, 2025) in direct generation mode and extended thinking at three budgets (4K, 8K, 16K tokens), yielding 4 configurations.

GPT-5-mini. We test GPT-5-mini (OpenAI, 2025) in direct generation mode and at three reasoning effort levels (4K, 8K, 16K tokens), yielding 4 configurations.

System Context

"You are an expert at solving NY Times Spelling Bee puzzles."

Available Letters (ONLY THESE 7)

`A | G | I | L | N | O | W`

CENTER LETTER (must be in every word): `O`

RULES

✓ Use ONLY these 7 letters: A, G, I, L, N, O, W
✓ Every word MUST contain the center letter: O
✓ Minimum 4 letters per word
✓ Letters can be reused (e.g., 'meet' uses E twice)
✓ Only common American English words
✓ BONUS: Words using all 7 letters (pangrams) are especially valuable

INSTRUCTIONS

1. Start with the center letter and build words around it
2. Focus on word patterns: common roots, prefixes, suffixes
4. Skip words with letters NOT in the available set
5. Find as many valid English words as possible
6. Only include words you are confident exist in standard dictionaries

OUTPUT FORMAT

After your thinking, provide ONLY a clean list of valid words.
- One word per line
- No numbers, bullets, or punctuation
- No explanations, notes, or commentary
- No blank lines between words

Start your word list now:

Figure 1: Zero-shot prompt structure. The prompt specifies the seven available letters, marks the mandatory center letter, and enumerates all constraints explicitly. Models receive identical specifications without solution counts, isolating intrinsic constraint-handling from memorization or calibration to expected output lengths.

3.5. Prompt Design

Figure 1 illustrates the zero-shot prompt structure. All models receive identical prompts specifying the seven available letters, mandatory center letter, and three constraints. Prompts use a target-free formulation ("Find as many valid English words as possible") without revealing solution counts, and request one word per line as output. For Qwen3, thinking mode automatically allocates reasoning tokens; Claude and GPT-5 handle reasoning internally.

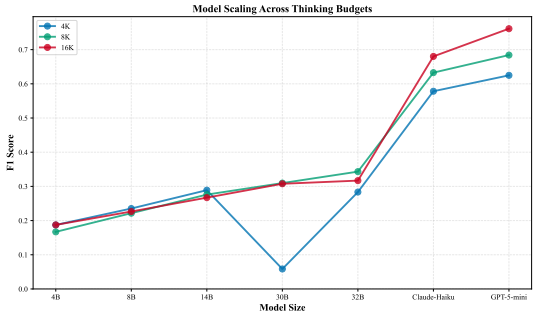

Figure 2: Cross-family performance comparison across thinking budgets. Proprietary models achieve 2.0–2.2× higher F1 than the largest open-source model, with the gap driven primarily by recall (68% vs. 23%) rather than precision. Budget sensitivity varies dramatically across families.

4. Evaluation Metrics

4.1. Standard Metrics

We evaluate generation quality using precision, recall, and F1 score. For instance i, let G_i denote generated words and S_i denote the verified solution set. Precision is $P = |G_i \cap S_i|/|G_i|$, recall is $R = |G_i \cap S_i|/|S_i|$, and F1 score is $F_1 = 2PR/(P + R)$.

4.2. Human Difficulty Alignment

Using solver data from 10,000 users per puzzle, we define human difficulty as $d_h(w) = 1 - (\text{user\_success\_count}/10{,}000)$, ranging from 0.03 to 0.97 across 2,007 words.

We compute two alignment metrics: (1) calibration strength, Pearson correlation between human difficulty and model difficulty (fraction of times the model missed each word), and (2) quartile stratification, model recall across human difficulty quartiles (Q1: easiest 25%, Q4: hardest 25%). Section 5.3 reveals modest but consistent calibration (r=0.24–0.38) alongside systematic model-specific failures.

4.3. Length-Stratified Analysis

We stratify recall by word length (4-letter, 5-letter, 6-letter, 7+ letters) to identify length-dependent failure patterns. Recall is computed as the fraction of words in each length range successfully generated, aggregated across all puzzles. Section 5.4 shows that models degrade catastrophically with word length (up to 82× for small models) while humans decline only 1.3×.

Model	Budget	P	R	F1	Vol
GPT-5-mini	16K	**0.888**	**0.680**	**0.761**	**34.4**
Claude-Haiku	16K	0.851	0.574	0.680	30.4
Claude-Haiku	8K	0.842	0.515	0.633	27.9
GPT-5-mini	8K	0.839	0.590	0.684	31.3
Qwen-32B	8K	0.797	0.233	0.343	12.7
Qwen-30B	8K	0.786	0.201	0.310	11.2
Qwen-14B	4K	0.768	0.184	0.289	10.6

Table 2: Top-performing configurations across model families (optimal budget per model). Performance gap manifests primarily through recall (68% vs. 23%) rather than precision (both 80–89%). All differences statistically significant ($p<0.001$). Vol is the volume of average word predictions.

5. Results

5.1. Architecture and Scale Effects on Constraint Satisfaction

Figure 2 reveals a performance hierarchy across model families. Proprietary models achieve F1 scores 2.0–2.2× higher than the largest open-source configuration we tested, with the cross-family gap substantially exceeding improvements from parameter scaling within the Qwen family. Such disparity between architectural and scale effects indicates that constraint satisfaction depends on factors beyond standard parameter scaling. The disproportionate benefit in proprietary models may reflect architectural innovations, specialized training objectives, or larger training datasets; we cannot disentangle these factors without access to proprietary details.

Precision-Recall Decomposition of Performance Differences Table 2 shows that the performance gap manifests primarily through recall rather than precision. GPT-5-mini achieves 68% recall compared to Qwen-32B's 23%, while precision increases by only ≈9%. This asymmetry has a specific implication: proprietary models discover more of the valid solution space rather than generating higher-quality guesses within the same search scope. The constraint satisfaction bottleneck lies in vocabulary coverage and search breadth, not in filtering invalid candidates. Rankings remain stable across all tested budgets, confirming fundamental capability gaps. Smaller proprietary models consistently outperform the largest open-source model tested, indicating these differences reflect core architectural or training factors rather than configuration-specific artifacts.

Universal Benefits of Extended Reasoning Table 3 shows that enabling thinking improves perfor-

Model	ΔP	ΔR	ΔF1	Rel. (%)
Qwen-4B	**+0.454**	+0.080	+0.134	+291
Qwen-8B	+0.302	+0.109	+0.172	+310
Qwen-14B	+0.301	**+0.140**	**+0.214**	**+335**
Qwen-30B	+0.155	+0.087	+0.130	+137
Qwen-32B	+0.187	+0.141	+0.197	+169

Table 3: Thinking mode effects averaged across budgets (Δ = Thinking ON − Thinking OFF). All models benefit from thinking, with gains manifesting primarily through precision. Relative gains inversely correlate with baseline performance.

mance across all tested models. Thinking mode provides universal benefits across model families, though the mechanism differs from conventional expectations. Counterintuitively, improvements manifest primarily through precision rather than recall: thinking mode reduces false positives more than it expands solution coverage. This pattern holds across the Qwen family and extends to Claude-Haiku, indicating that extended reasoning enables better constraint verification rather than broader search strategies. The varying relative gains reflect baseline performance differences: models with lower non-thinking baselines show larger percentage improvements, though absolute gains remain more consistent.

5.2. Model-Dependent Budget Sensitivity

Figure 3 illustrates heterogeneous budget trajectories across the Qwen3 family and proprietary models, revealing four distinct behavioral classes. Optimal thinking budgets vary dramatically across models, with some showing zero or negative returns from additional tokens.

Responsive High-Capacity Models: Dense and MoE Dynamics Within the Qwen3 family, budget responsiveness appears only in the high-capacity variants (30B and 32B), though their architectures differ fundamentally: the 30B employs mixture-of-experts (MoE) while the 32B uses a dense architecture.

The MoE variant exhibits particularly dramatic budget dependence, underperforming other family members at minimal allocation yet doubling performance in the 4K-to-8K range before plateauing. This sensitivity likely reflects MoE-specific dynamics: insufficient budget may prevent routing through enough expert pathways to solve the constraint satisfaction problem, while adequate allocation enables productive expert composition.

In contrast, the dense 32B model shows steadier gains with gentler diminishing returns across the

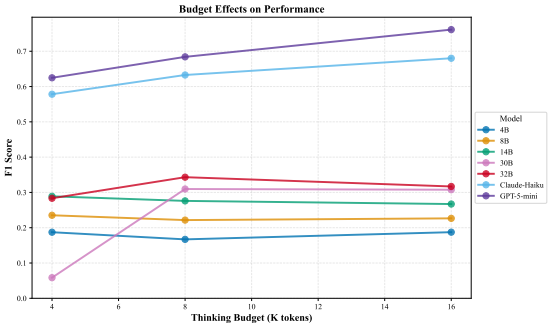

Figure 3: Heterogeneous budget sensitivity across model sizes. Smaller models remain flat or decline with additional budget, while the 14B variant paradoxically degrades. Proprietary systems show consistent improvements.

budget range. Both patterns reveal a counterintuitive deployment consideration: large models with insufficient budget may yield worse performance than smaller models at the same allocation, inverting typical parameter-scaling assumptions.

Budget-Insensitive Small Models The smallest variants (4B and 8B) exhibit budget insensitivity, maintaining flat performance across the tested range. Rather than engaging in deeper reasoning, these models generate redundant attempts or circular exploration. This pattern suggests a capacity threshold below which extended thinking becomes unproductive: these models may lack the representational capacity to decompose the constraint satisfaction problem into profitable sub-steps. Additional budget allocation provides no benefit when models cannot leverage it for meaningful computational progress.

Paradoxical Degradation in Mid-Sized Models The 14B variant shows a puzzling trend: performance worsens as reasoning budget increases. This may stem from a distributional mismatch: if the model was trained on shorter reasoning traces, extended thinking pushes it into unfamiliar regions where output quality declines. The result challenges the assumption that more reasoning always improves performance.

Proprietary Model Budget Efficiency Proprietary models (Claude-Haiku, GPT-5-mini) exhibit qualitatively different budget dynamics, showing strong, consistent positive returns across the full budget range without saturation or degradation. GPT-5-mini demonstrates the steepest efficiency gains, maintaining improvements from 4K through 16K tokens; Claude-Haiku follows a similar trajectory. These patterns contrast sharply with the Qwen family, where even responsive models

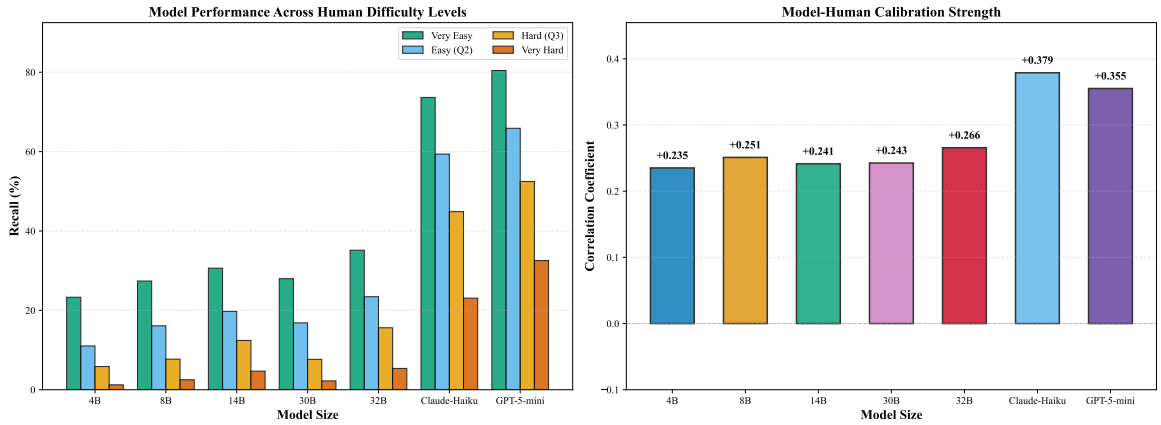

Figure 4: Model-human difficulty calibration using 10,000 solver ratings per puzzle. Left: Performance gradients from easy to hard words vary by model capacity (19× drop for Qwen-4B vs. 2.5× for GPT-5-mini). Right: Calibration strength shows modest alignment (r=0.24–0.38), with proprietary models achieving higher correlations.

plateau or decline at higher budgets. The consistent gains suggest that proprietary models may employ architectural features or training procedures that enable more effective utilization of extended reasoning resources. These model-specific dynamics raise a natural question: do models that perform better overall also align better with human difficulty perception? We examine this relationship using ground-truth solver data.

5.3. Human Difficulty Alignment with Model-Specific Gradients

Figure 4 presents systematic comparison between model and human difficulty using ground-truth performance data from 10,000 NYT Spelling Bee solvers. The analysis reveals shared patterns alongside persistent model-specific limitations.

Calibration Strength Across Model Families The Qwen family demonstrates positive correlations with human difficulty (r=0.235 to 0.266, all $p<0.001$), confirming that these models and humans share substantial overlap in which words prove challenging. Proprietary models exhibit moderately stronger calibration (Claude-Haiku r=0.38, GPT-5-mini r=0.36), though still far from perfect alignment. The open-source plateau around $r\approx0.25$ may reflect capacity limitations: models with insufficient representational power struggle to capture subtle orthographic features that humans use to judge difficulty. That proprietary models achieve higher correlations alongside superior absolute performance shows a correlation between model capability and human-like difficulty perception, though underlying drivers remain unclear.

Difficulty Gradients and Uniform Performance Gaps Model capacity dramatically affects difficulty gradients across human-defined difficulty quartiles. As shown in the left panel of Figure 4, smaller models show steep performance drops from easy to hard words (Qwen-4B: 19× drop), while larger open-source models maintain moderately better consistency (Qwen-32B: 6.5× drop). Proprietary models exhibit qualitatively flatter gradients (GPT-5-mini: 2.5× drop, Claude-Haiku: 3.2× drop). This pattern reveals that scaling within the Qwen family improves robustness to difficult words but does not eliminate the steep gradient phenomenon. The flatter proprietary model gradients suggest qualitatively different constraint-handling mechanisms that maintain performance even on vocabulary humans struggle with, contrasting sharply with smaller models that nearly collapse on hard words.

The proprietary models' absolute recall advantage persists uniformly across all difficulty quartiles. Even on the easiest words (Q1), GPT-5-mini achieves 80% recall compared to Qwen-32B's 35%, indicating that the performance gap cannot be explained solely by differential handling of difficult cases. This uniform advantage across the entire difficulty spectrum suggests that proprietary models possess more effective constraint-satisfaction capabilities, whether through architectural features, training data advantages, or other factors, rather than merely better strategies for edge cases or challenging vocabulary.

Systematic Failures and the Two-Component Difficulty Model Even the strongest correlations ($r\approx0.38$) leave substantial unexplained variance, revealing persistent model-specific failure modes. Models systematically miss common high-

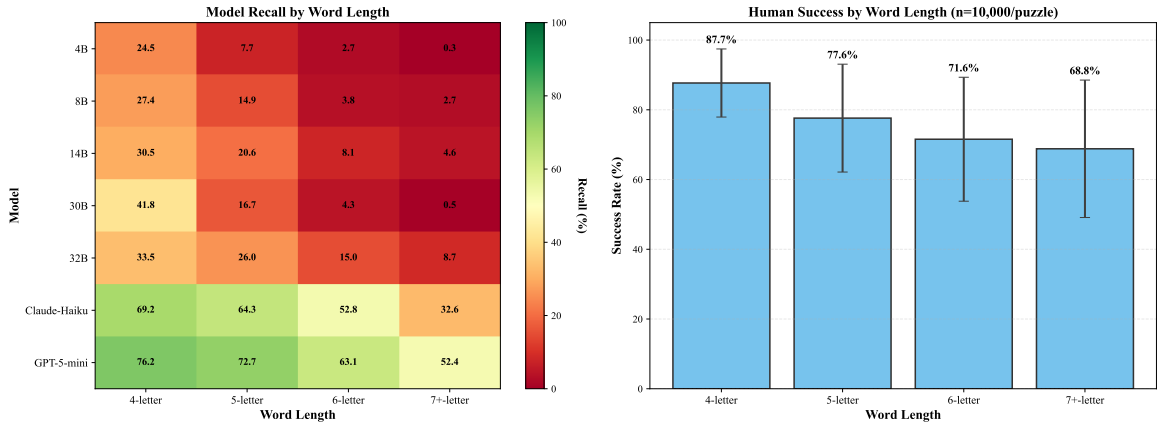

Figure 5: Word length effects on model and human performance. Left: Model recall by word length. Right: Human success declines gently (1.3× drop) while models show catastrophic degradation (1.5–82× drops).

Word	Model Miss (%)	Human Success (%)	Puzzles	Pattern
innie	100.0	81.0	3	Doubled cons.
illicit	97.0	86.7	3	Doubled cons.
loll	96.8	84.7	5	Doubled cons.
papa	95.5	88.8	3	Repeated letters
nana	95.5	86.4	4	Repeated letters
momma	95.5	87.2	3	Doubled cons.
acai	95.1	86.0	7	Truncated
toon	89.2	94.4	4	Repeated letters
poop	89.1	93.1	5	Repeated letters
data	89.1	93.4	5	Truncated

Table 4: Systematic failures: words with high human success (>80%) but high model miss rates (>89%) across all configurations. Puzzles indicates how many of the 58 puzzles contained each word. Pattern categories: doubled consonants (ll, nn, mm), repeated letters (symmetric patterns), truncated forms (4-letter abbreviated vocabulary).

frequency words that humans find trivial: “data” (93% human success, 89% model miss rate), “poop” (93% human, 89% model miss), and “toon” (94% human, 89% model miss) exemplify patterns where constraint-checking mechanisms fail despite vocabulary presence. These misses cannot be attributed to lexical gaps (all words appear frequently in pre-training corpora) but rather to failures in recognizing valid letter combinations under the specific constraint structure of each puzzle.

The moderate calibration (r=0.24–0.38) coupled with substantial model-specific failures is consistent with a two-component difficulty structure: (1) inherent linguistic difficulty that affects humans and models similarly (captured by correlations in the 0.2–0.4 range), and (2) model-specific difficulty arising from architectural, capacity, or training limitations (contributing the remaining 60–80% of variance). Proprietary models achieve both higher correlations and better absolute performance, indicating partial alignment between these components, though we cannot determine which specific factors drive this relationship. Substantial unexplained variance remains even for top models, suggesting important differences between how models represent orthographic constraints and how humans intuitively parse them, a challenge extending beyond simple parameter scaling or budget increases.

5.4. Length-Dependent Performance and Systematic Difficulty Patterns

Performance varies systematically based on word characteristics, revealing non-random failure modes. We examine two complementary perspectives: categorical orthographic patterns that trigger systematic failures, and continuous length-dependent degradation that affects all models to varying degrees.

Categorical Orthographic Pattern Failures Table 4 identifies words that humans find easy yet get missed by models at extraordinarily high rates across all configurations. When models fail, their misses cluster around specific orthographic patterns. These systematic failures reveal three difficulty categories: (1) doubled consonants, consecutive identical consonants like ll, nn, mm (“illicit,” “loll,” “innie,” “momma”), (2) repeated letters, words where the same letter appears multiple times non-consecutively, creating palindromic or symmetric patterns (“papa,” “nana,” “poop”), and (3) truncated forms, short 4-letter words often representing abbreviated or informal vocabulary (“acai,” “data,” “toon”).

Words like “data” and “poop” achieve high human success yet get consistently missed, despite appearing frequently in training corpora. These failures indicate systematic blind spots in constraint verification mechanisms that persist across model families and scales.

Length-Dependent Performance Degradation Figure 5 quantifies length-dependent patterns through direct model-human comparison. Performance stratified by word length reveals a consistent monotonic pattern: all models perform best on short words and progressively worse as length increases. This length gradient appears universal across model families but varies dramatically in magnitude, with weaker models collapsing almost entirely on longer words while stronger models maintain substantial recall even at length extremes. Both humans and models exhibit monotonic decline as word length increases, confirming that length inherently increases difficulty. However, the magnitude of degradation differs dramatically: human success declines gently from 87.7% (4-letter) to 68.8% (7+), a modest 1.3× reduction, while models show catastrophic drops ranging from 1.5× (GPT-5-mini) to 82× (Qwen-4B). This disparity reveals that while verification complexity affects both humans and models, neural architectures lack the robustness mechanisms humans employ to maintain performance as combinatorial complexity grows.

The differential length robustness contributes substantially to cross-family performance gaps. GPT-5-mini's 1.5× degradation approaches human-level robustness (1.3×), while Qwen-4B's 82× collapse reveals catastrophic failure of constraint-tracking. This difference is an aspect of model scaling but could also reflect architectural features, training differences, or other factors in proprietary models that enable more robust handling of complex multi-constraint verification.

Mechanistic Bottlenecks: Distributional Plausibility and Working Memory The observed failure patterns reveal two distinct computational bottlenecks that limit model performance on constraint satisfaction tasks.

Over-reliance on distributional plausibility. The categorical pattern failures reveal that constraint satisfaction engages different cognitive mechanisms than natural language generation. In free-form text, words like "data" and "papa" appear in predictable semantic contexts that prime their generation. In constraint satisfaction tasks, models must verify character-level properties independently of contextual cues. The near-universal failures on doubled consonants (consecutive ll in "illicit," nn in "innie," mm in "momma") and symmetric repeated patterns ("papa" with alternating p-a-p-a, "poop" with p-o-o-p) demonstrate that models rely heavily on distributional plausibility. These orthographically atypical but constraint-valid patterns trigger lower generation probabilities than more common letter sequences, even when both satisfy constraints equally. This explains why common words like "data" get systematically missed: the repeated 'a' creates a d-a-t-a pattern that appears distributionally implausible despite being a high-frequency English word.

Limited working memory for multi-constraint verification. The monotonic length-dependent decline likely reflects increasing verification complexity: longer words require checking more character positions, maintaining more constraint interactions simultaneously, and exploring larger combinatorial spaces. A 4-letter word involves 4 character checks and (4 choose 2) = 6 pairwise interactions; a 9-letter word involves 9 checks and 36 pairwise interactions. This quadratic growth in verification complexity affects both humans and models, yet humans maintain 69% success on 7+ letter words while small models collapse to 0.3%. Smaller Neural architectures lack the working memory capacity or systematic enumeration strategies humans employ.

Implications. Length-dependent patterns and systematic orthographic difficulties collectively reveal that constraint satisfaction failures are non-random. They cluster around structural properties (increasing length, unusual orthography) that expose specific computational bottlenecks: limited working memory for tracking multiple simultaneous constraints, over-reliance on distributional plausibility that penalizes valid-but-unusual patterns, and difficulty with systematic enumeration strategies. These failure modes suggest potential directions for improvement: explicit constraint-tracking mechanisms, training objectives that reward character-level verification independent of semantic plausibility, or prompting strategies that decompose long-word generation into systematic prefix enumeration.

6. Conclusion

Systematic evaluation across 28 configurations reveals that constraint satisfaction performance depends on factors beyond parameter scaling. Cross-family differences (2.0–2.2$\times$) substantially exceed within-family scaling gains (83%), manifesting through recall rather than precision. Thinking budget effects prove heterogeneous: the 14B model degrades with increased allocation while the MoE 30B variant requires substantial budget for productive expert composition.

Models systematically miss common words with atypical orthography ("data", "poop": 89% model miss vs. 93% human success), revealing persistent blind spots despite vocabulary knowledge. Calibration with 10,000 human solvers is modest (r=0.24–0.38), with substantial unexplained variance indicating fundamental differences in how models and humans represent orthographic constraints.

These findings suggest three research directions: (1) explicit verification modules operating independently of distributional priors, (2) training objectives rewarding orthographically unusual but valid solutions, and (3) model-specific budget allocation policies accounting for heterogeneous sensitivity. The core bottleneck we identify, over-reliance on distributional plausibility when structural validity is required, extends beyond word puzzles. Code completion must satisfy syntactic constraints regardless of token frequency; structured data generation requires schema conformance independent of common patterns; mathematical reasoning demands logical validity even when intermediate steps appear unlikely. In each case, correct outputs may be distributionally atypical, precisely the failure mode our analysis exposes.

7. Limitations

The experimental paradigm focuses exclusively on English orthography, limiting cross-linguistic generalization, though the distributional plausibility mechanism we identify may transfer to other alphabetic systems. The task emphasizes lexical retrieval and constraint satisfaction without requiring semantic understanding, isolating one component of constrained generation while leaving semantic constraint interactions unexplored. Zero-shot evaluation reveals intrinsic constraint-handling capabilities; few-shot prompting might mitigate some failures but would not address underlying architectural limitations. The 58-puzzle dataset provides statistical power for configuration comparisons (1,624 experiments across 2,007 unique words) but limits analysis of temporal patterns or rare letter combinations.

8. Ethical Considerations

This work evaluates models on a publicly available word generation task, raising minimal ethical concerns. The experimental instances are drawn from the New York Times Spelling Bee for research and evaluation purposes consistent with fair use principles. We do not redistribute puzzle content beyond what is necessary for reproducibility. The word lists reflect editorial curation choices and may encode vocabulary biases, though the task focuses on orthographic constraints rather than semantic content.

Acknowledgments

Research partly supported by NSF grant 2244279, ARO grant W911NF-23-1-0191 and a US Department of Transportation grant for CyberCare. Verma is the founder of Everest Cyber Security and Analytics, Inc.

9. References

Jacob Andreas. 2022. Language models as agent models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 5769–5779, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Anthropic. 2025. Claude opus 4 & claude sonnet 4 system card. Technical report, Anthropic. Models: Claude Opus 4, Claude Sonnet 4.

Lukas Beurer-Kellner et al. 2024. Guiding llms the right way: Fast, non-invasive constrained decoding. *arXiv:2403.06988*.

Bastian Bunzeck, Daniel Duran, Leonie Schade, and Sina Zarrieß. 2024. Graphemes vs. phonemes: battling it out in character-based language models. In *The 2nd BabyLM Challenge at the 28th Conference on Computational Natural Language Learning*, pages 54–64, Miami, FL, USA. Association for Computational Linguistics.

Yekun Chai, Yewei Fang, Qiwei Peng, and Xuhong Li. 2024. Tokenization falling short: On subword robustness in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1582–1599, Miami, Florida, USA. Association for Computational Linguistics.

Thao Anh Dang, Limor Raviv, and Lukas Galke. 2025. Tokenization and morphology in multilingual language models: A comparative analysis of mT5 and ByT5. In *Proceedings of the 8th International Conference on Natural Language and Speech Processing (ICNLSP-2025)*, pages 242–257, Southern Denmark University, Odense, Denmark. Association for Computational Linguistics.

Nouha Dziri, Ximing Lu, Melanie Sclar, Xiang Lorraine Li, Liwei Jiang, Bill Yuchen Lin, Peter West, Chandra Bhagavatula, Ronan Le Bras, Jena D. Hwang, Soumya Sanyal, Sean Welleck, Xiang Ren, Allyson Ettinger, Zaid Harchaoui, and Yejin Choi. 2023. Faith and fate: Limits of transformers on compositionality.

Lukas Edman, Helmut Schmid, and Alexander Fraser. 2024. CUTE: Measuring LLMs' understanding of their tokens. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3017–3026,

Miami, Florida, USA. Association for Computational Linguistics.

Cristina Garbacea and Qiaozhu Mei. 2025. Why is constrained neural language generation particularly challenging? *Transactions on Machine Learning Research*.

Panagiotis Giadikiaroglou, Maria Lymperaiou, Giorgos Filandrianos, and Giorgos Stamou. 2024. Puzzle solving using reasoning of large language models: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 11574–11591, Miami, Florida, USA. Association for Computational Linguistics.

Chris Hokamp and Qun Liu. 2017. Lexically constrained decoding for sequence generation using grid beam search. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1535–1546, Vancouver, Canada. Association for Computational Linguistics.

Inbal Itzhak, Ran Zmigrod, and Ryan Cotterell. 2022. Models in a spelling bee: Language models implicitly learn to spell. In *NAACL*.

Abdelhak Kelious, Mathieu Constant, and Christophe Coeur. 2024. Complex word identification: A comparative study between ChatGPT and a dedicated model for this task. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 3645–3653, Torino, Italia. ELRA and ICCL.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. 2023. Transformers learn shortcuts to automata.

Ximing Lu, Sean Welleck, Peter West, Liwei Jiang, Jungo Kasai, Daniel Khashabi, Ronan Le Bras, Lianhui Qin, Youngjae Yu, Rowan Zellers, Noah A. Smith, and Yejin Choi. 2022. NeuroLogic a*esque decoding: Constrained text generation with lookahead heuristics. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 780–799, Seattle, United States. Association for Computational Linguistics.

Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. 2023. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Comput. Surv.*, 56(2).

Adam Nohejl, Frederikus Hudi, Eunike Andriani Kardinata, Shintaro Ozaki, Maria Angelica Riera Machin, Hongyu Sun, Justin Vasselli, and Taro Watanabe. 2025. Beyond film subtitles: Is YouTube the best approximation of spoken vocabulary? In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9566–9585, Abu Dhabi, UAE. Association for Computational Linguistics.

OpenAI. 2025. Gpt-5 system card. Technical report, OpenAI. Version: GPT-5 (August 13 2025).

Kanghee Park, Timothy Zhou, and Loris D'Antoni. 2025. Flexible and efficient grammar-constrained decoding.

Friedemann Pulvermüller. 2010. Brain-language research: Where is the progress? *Biolinguistics*.

Federico Raspanti, Tanir Ozcelebi, and Mike Holenderski. 2025. Grammar-constrained decoding makes large language models better logical parsers. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 485–499, Vienna, Austria. Association for Computational Linguistics.

Soumadeep Saha, Sutanoya Chakraborty, Saptarshi Saha, and Utpal Garain. 2025. Language models are crossword solvers. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, page 2074–2090. Association for Computational Linguistics.

Ashima Suvarna, Harshita Khandelwal, and Nanyun Peng. 2024. Phonologybench: Evaluating phonological skills of large language models. In *ACL Workshop on Knowledge Extraction from LLMs*.

Eshaan Tanwar, Gayatri Oke, and Tanmoy Chakraborty. 2025. Multilingual llms struggle to link orthography and semantics in bilingual word processing. *arXiv:2501.09127*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu

Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. 2025. Qwen3 technical report.

Kayo Yin and Graham Neubig. 2022. Interpreting language models with contrastive explanations. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 184–198, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.