CTGBench: Constrained Text Generation for Language Models

Shiny Weng [1] **Lillian Weng** [1] **James Cheng** [1]

Abstract

While modern Large Language Models (LLMs) demonstrate remarkable semantic reasoning, their auto-regressive architecture imposes a fundamental limitation: a "myopic" generation process that lacks inherent look-ahead planning. This deficiency becomes critical in *Constrained Text Generation (CTG)*, where satisfying complex constraints requires anticipating future tokens before they are generated. We introduce CTGBENCH, a benchmark designed to isolate and stress-test this planning gap in modern LLMs. CTGBENCH contains 348 parameterized prompt templates (101 local, 97 global, 150 hybrid) and 7,740 instantiated prompts. By deconstructing tasks into (i) *Local* (token-level), (ii) *Global* (sequence-level), and (iii) *Hybrid* (mixed token-level and sequence-level) constraints, we provide a precise diagnostic tool, including a robust evaluation suite, for quantifying and improving long-horizon controllability in language generation. Across five models, we find that structural control remains limited: no model exceeds 25.1% accuracy on hybrid constraints. Our code is made accessible via https://github.com/shinyweng/CTGBench.

1. Introduction

Auto-regressive language models generate text one token at a time, conditioning on the prefix they have already produced. This factorization is highly effective for open-ended generation, but it becomes brittle when validity depends on information that lies in the future of the sequence. A model asked to write a paragraph with exactly 150 characters, or a list whose final item must close a structural dependency introduced earlier, must commit to early token choices before it has "seen" the full output it must ultimately satisfy.

This challenge arises naturally in *Constrained Text Generation* (CTG), where correctness depends not only on semantic fluency but also on adherence to explicit structural rules. Formally, CTG can be viewed as generating a sequence of tokens $y = (t_1, \ldots, t_n)$ such that a constraint function $C(y)$ evaluates to true. When C depends on properties of the entire sequence—such as length, formatting, or cross-token dependencies—the optimal choice of early tokens may depend on decisions that occur much later in the generation process.

Consider the prompt: *Write five tips for young adults such that the last letters of each tip spell "F-R-U-I-T"*. To succeed, the model must choose the *first* words of early sentences while anticipating the *last* letters of words that will appear much later in the sequence. Despite strong language modeling ability, current systems frequently produce fluent outputs that violate such structural constraints.

We refer to this failure mode as *planning myopia*: the gap between locally coherent next-token prediction and globally valid constrained generation. Intuitively, the choice of the current token t_i may depend on properties of future tokens t_{i+n}—such as their length, content, or formatting—yet standard left-to-right decoding provides no intrinsic mechanism for reasoning over that future constraint space.

This failure mode is easy to miss in standard evaluations. Benchmarks such as MMLU (Hendrycks et al., 2020) and HumanEval (Chen et al., 2021) emphasize knowledge, reasoning, or executable correctness, but they do not isolate whether a model can satisfy strict structural constraints in natural-language generation. Instruction-following benchmarks such as IFEval (Zhou et al., 2023) are closer in spirit, yet they span many instruction types and do not center around auditable long-horizon constraint satisfaction as a first-class axis.

To study this capability directly, we introduce a taxonomy of constraint types that vary in the degree to which they require local versus global planning:

- **Local constraints:** depend only on the next token (e.g., "Every word must start with the letter 'A").
- **Global constraints:** depend on properties of the entire sequence (e.g., "Exactly 20 words total").
- **Hybrid constraints:** combine token-level and

[1]Department of Computer Science, Stanford University. Correspondence to: Shiny Weng <shinyweng@stanford.edu>, Lillian Weng <lillianweng@stanford.edu>, James Cheng <jamescheng@stanford.edu>.

CS 525, Data for AI

sequence-level requirements (e.g., "A 300-character movie review containing no letter 'e'").

Contributions. We make three contributions. First, we introduce CTGBENCH, a benchmark for constrained text generation consisting of 348 parameterized prompt templates (101 local, 97 global, and 150 hybrid) and 7,740 instantiated prompts. Second, we provide a robust evaluation pipeline that performs deterministic programmatic verification whenever possible and falls back to a strict LLM-based grader only for the small subset of prompts whose correctness cannot be automatically checked. Third, we present an empirical study across five models showing that structural control remains limited even for strong systems: on the shared evaluation split, no model exceeds 25.1% accuracy on hybrid constraints, and fluent outputs frequently fail simple structural requirements.

2. Related Work

While general-purpose benchmarks like MMLU (Hendrycks et al., 2020) and HumanEval (Chen et al., 2021) effectively evaluate broad reasoning and coding abilities, they do not assess *structural controllability* in natural language generation—specifically, a model's capacity to satisfy strict sequence-level constraints like exact counts, positional dependencies, or symbolic format rules.

Instruction-following benchmarks are more relevant to our setting. IFEval (Zhou et al., 2023) measures whether models satisfy explicit instructions, while FollowBench (Jiang et al., 2024) studies fine-grained constraint following. CTGBENCH differs in emphasis: we focus specifically on constraint families that expose planning requirements in left-to-right generation and pair them with auditable scoring whenever possible.

Our benchmark is also complementary to inference-time control methods. Grammar-constrained decoding (Geng et al., 2024), lexically constrained decoding (Hokamp & Liu, 2017), and lookahead heuristics such as NeuroLogic A*esque (Lu et al., 2022) aim to enforce structure during generation. CTGBENCH instead measures the *underlying* constrained-generation ability of a model before specialized decoding is applied, making it useful both as a standalone evaluation and as a target for future control methods.

3. CTGBENCH Design

3.1. Constraint Taxonomy

We define a taxonomy of constraints to systematically categorize tasks by increasing levels of difficulty and cognitive load. Formally, we define these as:

- **Local Constraints:** Constraints where the validity of a token t_i can be determined solely by the immediate context window $t_{i-k}, \ldots, t_{i-1}$, independent of the total sequence length or future states. Affects the next token only (e.g., "Start every word with 'A'").
- **Global Constraints:** Constraints applied to the aggregate properties of the complete sequence X. The validity function $f(X)$ returns True/False only upon the `EOS` (End of Sequence) token. Requires maintaining a running count or looking ahead (e.g., "Exactly 20 words total").
- **Hybrid Constraints:** A composition of one local and one global constraint.

3.2. Semantic Requirements (Orthogonal Axis)

Nearly all prompts include a topical requirement; we evaluate semantic adequacy orthogonally on a 1–3 scale so that structural satisfaction is not conflated with meaning preservation. We utilize our LLM grader to assess semantic coherence.

3.3. Prompt Construction and Split

CTGBENCH is defined in code as a parameterized prompt bank. The current version contains 101 local, 97 global, and 150 hybrid base templates. Prompt parameters such as topics, counts, target letters, and formatting symbols expand the 348 base templates into 7,740 concrete instances.

The shared evaluation split contains 97 base templates and 1,837 expanded prompts: 27 local, 32 global, and 38 hybrid base families, corresponding to 160, 675, and 1,002 concrete prompts respectively. We use this split for cross-model comparison because all five main models were scored on it under the same protocol.

Table 1. CTGBENCH statistics. "A/B" denotes exact programmatic versus LLM-judged template families.

Constraint	Base	A/B	Expanded	Eval
Local	101	96/5	760	160
Global	97	93/4	2,047	675
Hybrid	150	136/14	4,933	1,002
Total	348	325/23	7,740	1,837

3.4. Feasibility Auditing

All expanded prompts were audited for feasibility. We flagged and removed or revised prompts that were impossible or near-impossible to satisfy—for example, requiring every word in a long passage to have exactly five syllables while maintaining coherent prose on a specific topic. For hybrid prompts in particular, auditing verified that the two component constraints do not conflict (e.g., "every word

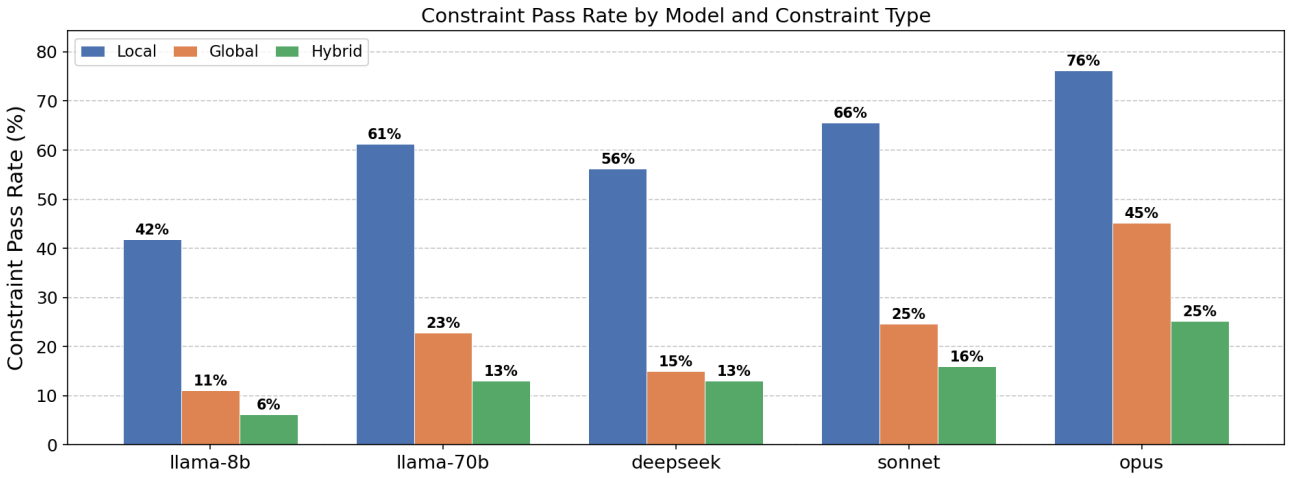

Figure 1. Constraint pass rate by model and constraint type on the shared evaluation split. The local > global > hybrid ordering holds for every model.

starts with 'S'" combined with "'and' appears exactly 3 times") or trivially subsume one another.

3.5. Verification Modes

Each base prompt family is assigned one of two verification modes in the repository:

- **A-type**: correctness is fully and deterministically checkable with a lightweight Python verifier.
- **B-type**: exact verification is not reliable with lightweight rules alone, typically because the prompt depends on semantics, lexical knowledge, or grammatical judgment. We utilize a Qwen3-235B-A22B-Instruct model (Yang et al., 2025) as the LLM-as-a-judge (Zheng et al., 2023).

The evaluation pipeline always computes programmatic verifier output when available and also records LLM-grader judgments. However, the final *authoritative* pass/fail label uses the verifier type attached to the prompt family: A-type prompts trust the exact verifier, and B-type prompts trust the LLM grader. On the shared evaluation split, 1,784 of 1,837 prompts (97.1%) are A-type.

This distinction matters in practice. When both signals are available on A-type evaluation prompts, programmatic and LLM judgments may disagree. We therefore treat exact verifiers as authoritative whenever possible.

4. Experimental Setup

We evaluate five instruction-tuned models: Llama-3.1-8B, Llama-3.3-70B (Grattafiori et al., 2024), DeepSeek-V3.1 (DeepSeek-AI et al., 2025), Claude Sonnet 4.6 (Anthropic, 2026b), and Claude Opus 4.6 (Anthropic, 2026a). All generations use the same system prompt, temperature 0.2, and a maximum of 512 new tokens. The system prompt requests that the final answer be enclosed in `<answer>` tags so that scoring can reliably isolate the model's final output.

Models were run on the full 7,740-instance benchmark, after which the shared evaluation split was extracted. However, Claude models were generated only on the same 1,837 evaluation prompts due to monetary constraints. For every response, we record authoritative pass/fail and an auxiliary semantic coherence score from 1 to 3, where 3 denotes coherent, relevant, and semantically sound output.

Table 2. Authoritative pass rates (%) on the shared 1,837-prompt evaluation split.

Model	Local	Global	Hybrid	Overall
Llama-3.1-8B	41.9	11.1	6.2	11.1
Llama-3.3-70B	61.3	22.8	13.1	20.8
DeepSeek-V3.1	56.2	15.0	13.1	17.5
Claude Sonnet 4.6	65.6	24.6	16.0	23.5
Claude Opus 4.6	76.2	45.2	25.1	37.0

5. Results

5.1. Main Benchmark

Table 2 reports exact authoritative pass rates on the shared evaluation split, and Figure 1 visualizes the per-type pattern.

Three findings stand out: first, all models exhibit an expected difficulty gradient from local to global to hybrid constraints. Averaged across models, pass rate drops from 60.2% on local prompts to 23.7% on global prompts and 14.7% on hybrid prompts. Second, scale helps but does not

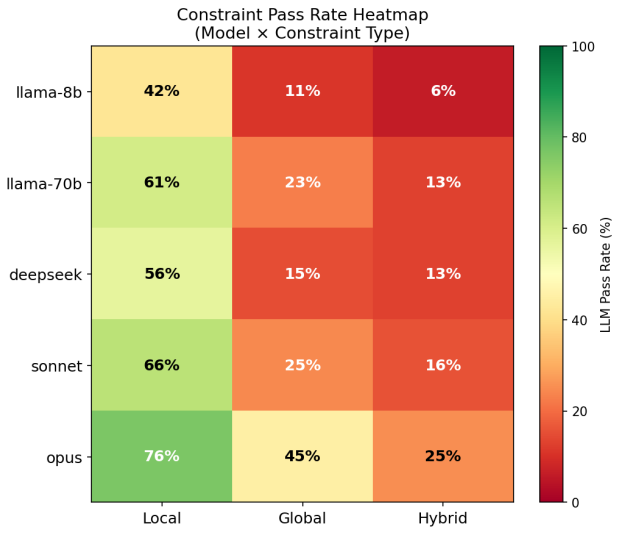

Figure 2. Constraint pass rate heatmap across models and constraint types. Performance degrades consistently from local to global to hybrid constraints, with Claude Opus 4.6 performing best overall but still showing substantial failure on hybrid prompts.

solve the problem: Claude Opus 4.6 more than triples the overall score of Llama-3.1-8B (37.0% vs. 11.1%), yet still fails 63.0% of evaluation prompts (Figure 2). Third, hybrid prompts remain especially challenging. No model exceeds 25.1% accuracy despite the fact that hybrid prompts are constructed from component constraints, local and global, that are individually solvable at much higher rates.

5.2. Semantic Fluency Does Not Imply Structural Correctness

The benchmark separates semantic adequacy from structural validity (Figure 3). Although passing results often do have higher semantic coherence scores, across all failed evaluation outputs with available semantic ratings, the mean semantic coherence score is still high at 2.42/3. For successful outputs it is 2.90/3. In other words, many failures are not nonsense; they are fluent, topical answers that miss the constraint by a small but decisive margin. For example, Claude Opus 4.6 produces the fluent sentence "Yearly supply naturally flows slowly, steadily, orderly, downwardly" for a prompt requiring every word to end in `-ly`; this response fails on the single word `flows`. Appendix A.2 contains additional examples.

5.3. Fine-tuning Results

Table 3 and Figure 4 report the effect of supervised fine-tuning (SFT) with Low-Rank Adaptation (LoRA) (Hu et al., 2021) on the two open-weight Llama models. Fine-tuning data was constructed from the training split using the authoritative verifier for each constraint (programmatic for A-type, LLM-graded for B-type) as reward signals: for each training prompt, we sampled multiple completions and retained only those that passed all constraints.

Table 3. Effect of SFT on constraint pass rates (%). Δ denotes the absolute change from the base model.

Model	Variant	Local	Global	Hybrid	Overall
Llama-3.1-8B	Base	41.9	11.1	6.2	11.1
	SFT	61.3	14.4	9.1	15.6
	Δ	+19.4	+3.3	+2.9	+4.5
Llama-3.3-70B	Base	61.3	22.8	13.1	20.8
	SFT	67.5	21.5	12.5	20.6
	Δ	+6.2	-1.3	-0.6	-0.2

SFT produces a striking asymmetry across constraint types. For Llama-3.1-8B, fine-tuning improves all categories, with local constraints seeing the largest gain (+19.4 percentage points). Llama-3.3-70B shows a similar local improvement (+6.2 pp) but *regresses* on global (-1.3 pp) and hybrid (-0.6 pp) constraints, resulting in a near-zero overall change (-0.2 pp).

This pattern suggests that SFT effectively teaches surface-level patterns—formatting rules, word-level constraints, and other token-local properties that can be learned from input–output pairs. However, global constraints (e.g., maintaining a running word count, structuring a multi-paragraph argument, or satisfying cross-sentence dependencies) require planning and compositional reasoning that outcome-based SFT does not directly optimize. For Llama-3.3-70B, which already handles many local constraints in its base form, SFT appears to overfit to local patterns at the expense of global coherence. These findings motivate future work on process reward models and search-time methods that can provide fine-grained feedback on intermediate generation steps.

6. Limitations and Future Work

CTGBENCH evaluates single-turn, text-only generation and does not cover multi-turn interaction, tool use, or multimodal constraints. Our taxonomy, while systematic, is not exhaustive: real-world CTG tasks may involve domain-specific rules or soft preferences that resist binary verification. The B-type (LLM-judged) subset, though small (2.9% of eval prompts), introduces grader variance. Our fine-tuning experiments use LoRA-based SFT only; other approaches such as DPO (Rafailov et al., 2023) or RLHF (Ouyang et al., 2022) may yield different patterns.

These gaps suggest several directions. First, *Structural Chain-of-Thought*: training models to maintain intermediate state variables—running counts, letter queues, dependency structures—in a scratchpad before producing the final output, analogous to how CoT improves mathematical reason-

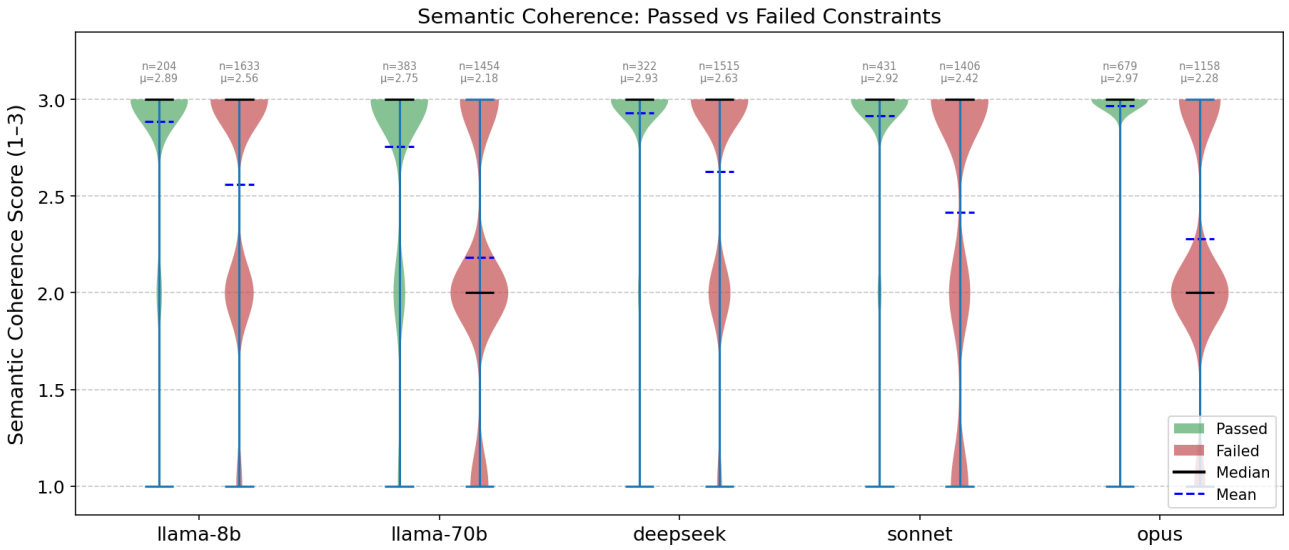

Figure 3. Distribution of semantic coherence scores for passed versus failed outputs across models. Even failed generations are frequently coherent and relevant, indicating that semantic fluency is a weak proxy for exact constraint satisfaction.

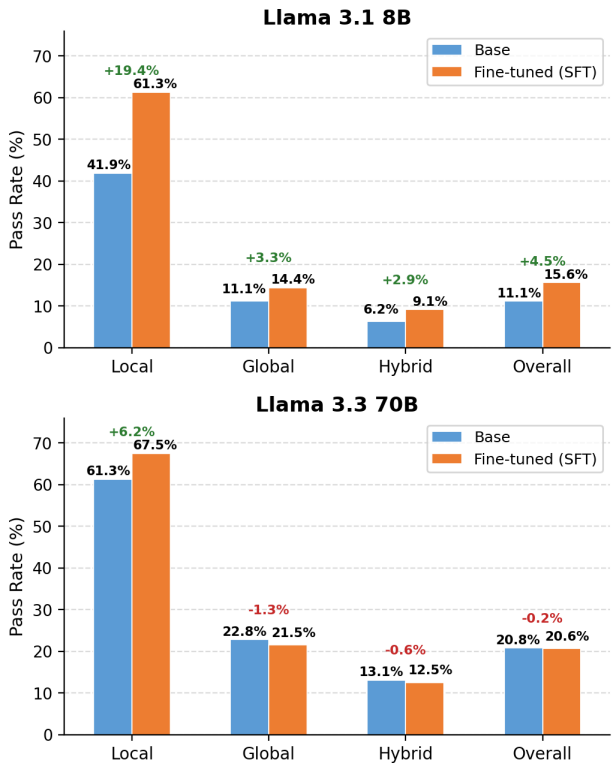

Figure 4. SFT impact on constraint pass rates. Fine-tuning yields large gains on local (token-level) constraints for both models, but fails to improve—and slightly degrades—global and hybrid performance for the larger 70B model.

ing (Wei et al., 2022). Second, *process-level supervision*: Process Reward Models (Lightman et al., 2023) that evaluate the viability of partial sequences against global constraints, enabling models to backtrack before a constraint is irrevocably violated. Finally, expanding CTGBENCH to multilingual, structured data (e.g., complex JSON schemas), and multimodal settings would test whether planning myopia generalizes beyond English text generation.

7. Conclusion

We present CTGBENCH, a benchmark for measuring constrained text generation under local, global, and hybrid structural requirements. Across five models, strong semantic fluency coexists with poor structural control, especially once prompts require long-horizon planning or the conjunction of multiple constraints. We hope CTGBENCH serves as a useful target for future work on controllable generation, verifier design, post-training, and constrained decoding.

References

Anthropic. Introducing claude opus 4.6. https://www.anthropic.com/news/claude-opus-4-6, 2026a. Anthropic News.

Anthropic. Introducing claude sonnet 4.6. https://www.anthropic.com/news/claude-sonnet-4-6, 2026b. Anthropic News.

Chen, M., Tworek, J., Jun, H., Yuan, Q., de Oliveira Pinto, H. P., Kaplan, J., Edwards, H., Burda, Y., Joseph, N., Brockman, G., Ray, A., Puri, R., Krueger, G., Petrov, M., Khlaaf, H., Sastry, G., Mishkin, P., Chan, B., Gray,

S., Ryder, N., Pavlov, M., Power, A., Kaiser, L., Bavarian, M., Winter, C., Tillet, P., Such, F. P., Cummings, D., Plappert, M., Chantzis, F., Barnes, E., Herbert-Voss, A., Guss, W. H., Nichol, A., Paino, A., Tezak, N., Tang, J., Babuschkin, I., Balaji, S., Jain, S., Saunders, W., Hesse, C., Carr, A. N., Leike, J., Achiam, J., Misra, V., Morikawa, E., Radford, A., Knight, M., Brundage, M., Murati, M., Mayer, K., Welinder, P., McGrew, B., Amodei, D., McCandlish, S., Sutskever, I., and Zaremba, W. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL https://arxiv.org/abs/2107.03374.

DeepSeek-AI, Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., Yang, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Zhang, H., Ding, H., Xin, H., Gao, H., Li, H., Qu, H., Cai, J. L., Liang, J., Guo, J., Ni, J., Li, J., Wang, J., Chen, J., Chen, J., Yuan, J., Qiu, J., Li, J., Song, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Xu, L., Xia, L., Zhao, L., Wang, L., Zhang, L., Li, M., Wang, M., Zhang, M., Zhang, M., Tang, M., Li, M., Tian, N., Huang, P., Wang, P., Zhang, P., Wang, Q., Zhu, Q., Chen, Q., Du, Q., Chen, R. J., Jin, R. L., Ge, R., Zhang, R., Pan, R., Wang, R., Xu, R., Zhang, R., Chen, R., Li, S. S., Lu, S., Zhou, S., Chen, S., Wu, S., Ye, S., Ye, S., Ma, S., Wang, S., Zhou, S., Yu, S., Zhou, S., Pan, S., Wang, T., Yun, T., Pei, T., Sun, T., Xiao, W. L., Zeng, W., Zhao, W., An, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Li, X. Q., Jin, X., Wang, X., Bi, X., Liu, X., Wang, X., Shen, X., Chen, X., Zhang, X., Chen, X., Nie, X., Sun, X., Wang, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yu, X., Song, X., Shan, X., Zhou, X., Yang, X., Li, X., Su, X., Lin, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhu, Y. X., Zhang, Y., Xu, Y., Xu, Y., Huang, Y., Li, Y., Zhao, Y., Sun, Y., Li, Y., Wang, Y., Yu, Y., Zheng, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Tang, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Wu, Y., Ou, Y., Zhu, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Zha, Y., Xiong, Y., Ma, Y., Yan, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Wu, Z. F., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Huang, Z., Zhang, Z., Xie, Z., Zhang, Z., Hao, Z., Gou, Z., Ma, Z., Yan, Z., Shao, Z., Xu, Z., Wu, Z., Zhang, Z., Li, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Gao, Z., and Pan, Z. Deepseek-v3 technical report, 2025. URL https://arxiv.org/abs/2412.19437.

Geng, S., Josifoski, M., Peyrard, M., and West, R. Grammar-constrained decoding for structured nlp tasks without finetuning, 2024. URL https://arxiv.org/abs/2305.13971.

Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poul-

ton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S., Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. *CoRR*, abs/2009.03300, 2020. URL https://arxiv.org/abs/2009.03300.

Hokamp, C. and Liu, Q. Lexically constrained decoding for sequence generation using grid beam search, 2017. URL https://arxiv.org/abs/1704.07138.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. Lora: Low-rank adaptation of large language models, 2021. URL https://arxiv.org/abs/2106.09685.

Jiang, Y., Wang, Y., Zeng, X., Zhong, W., Li, L., Mi, F., Shang, L., Jiang, X., Liu, Q., and Wang, W. Followbench: A multi-level fine-grained constraints following benchmark for large language models, 2024. URL https://arxiv.org/abs/2310.20410.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Lu, X., Welleck, S., West, P., Jiang, L., Kasai, J., Khashabi, D., Bras, R. L., Qin, L., Yu, Y., Zellers, R., Smith, N. A., and Choi, Y. Neurologic a*esque decoding: Constrained text generation with lookahead heuristics, 2022. URL https://arxiv.org/abs/2112.08726.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C. L., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., Schulman, J., Hilton, J., Kelton, F., Miller, L., Simens, M., Askell, A., Welinder, P., Christiano, P., Leike, J., and Lowe, R. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D., and Finn, C. Direct preference optimization: Your language model is secretly a reward model, 2023. URL https://arxiv.org/abs/2305.18290.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models, 2022. URL https://arxiv.org/abs/2201.11903.

Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. URL https://arxiv.org/abs/2306.05685.

Zhou, J., Lu, T., Mishra, S., Brahma, S., Basu, S., Luan, Y., Zhou, D., and Hou, L. Instruction-following evaluation for large language models, 2023. URL https://arxiv.org/abs/2311.07911.

A. Appendix

A.1. Example of Prompts Across Constraints

Table 4. **Taxonomy of Constrained Text Generation Tasks.** We categorize constraints into three levels of planning complexity: *Local* (token-level), *Global* (sequence-level), and *Hybrid* (multi-objective). The table provides representative examples from our proposed dataset used to evaluate auto-regressive planning capabilities.

Constraint Class	Example Prompt (Input)
Local Constraints *(Token-Level)*	1. Explain how a car engine works using only words that have exactly one syllable. 2. Write a sentence where each word is strictly longer than the word immediately preceding it. 3. Describe what a forest is without using the letter 'e'.
Global Constraints *(Sequence-Level)*	1. Write a review of a movie using exactly 20 words. The total number of the letter 'e' used in the entire review must be exactly 4. 2. Write a poem about space where the first letters of each line spell out the word 'GALAXY'. 3. Write a text where the word 'time' appears exactly at the beginning, the exact middle, and the very end of the sequence.
Hybrid Constraints *(Local + Global)*	1. Write a review of a book using exactly 15 occurrences of the letter "a". Additionally, every single word must contain at least one set of double letters (e.g., book, apple, seed). 2. Write a 5-sentence paragraph about classical music where the total word count is exactly 80. Additionally, every single word must start with the letter "S". 3. Write a short story about the ocean where the first and last words are identical. Additionally, every single word must start with the letter "S".

A.2. Example of Output & Failures

Table 5. **Qualitative Examples of Output & Failures.** We highlight representative failures where auto-regressive models generate semantically coherent text that violates structural constraints.

Constraint Type	Prompt & Model Response	Failure Analysis
Local	**Prompt:** Describe what a forest is without using the letter 'e'. **Output:** A forest is a vast natural world full of tall **tre**es and **de**nse undergrowth...	**Failure:** Uses the letter 'e' in "trees" and "dense". The model produces fluent prose but fails the token-level constraint.
Global	**Prompt:** Write a review of a movie using exactly 20 words. The total number of the letter 'e' used in the entire review must be exactly 3. **Output:** A thrilling film, full of twists. It's a grand story, told with skill.	**Failure:** Expected exactly 20 words, got 13. The model fails both the word count and letter count constraints.
Hybrid	**Prompt:** Write a 5-sentence paragraph about classical music where the total word count is exactly 80. Additionally, every word must start with the letter 'S'. **Output:** Symphonic sounds soar softly, stirring souls since centuries started...	**Failure:** Maintains the per-word constraint (S-words) but fails the exact word count target.
Hybrid	**Prompt:** Write a story about a lost key that is exactly 100 words long. Additionally, use only words with exactly one syllable. **Output:** Yearly supply naturally flows slowly, steadily, orderly, downwardly...	**Failure:** Multi-syllable words like "naturally" and "steadily" violate the one-syllable local constraint.