Beyond Leaderboards: Tokenomics of Agentic Small Language Model Ensembles

Alexei N. Skurikhin
alexei@lanl.gov
Los Alamos National Laboratory
Los Alamos, New Mexico, USA

Emily M. Taylor
ecasleton@lanl.gov
Los Alamos National Laboratory
Los Alamos, New Mexico, USA

Nathan A. DeBardeleben
ndebard@lanl.gov
Los Alamos National Laboratory
Los Alamos, New Mexico, USA

Abstract

As large language models (LLMs) move from standalone assistants into agentic workflows, evaluation must extend beyond scalar leaderboard accuracy to account for operational reliability, cost, latency, and token efficiency. We use an agentic ensemble of small language models (SLMs) with an SLM-judge-mediated feedback loop as a case study for such beyond-leaderboard evaluation. On the 541-prompt IFEval benchmark, the best ensemble achieves 97.34% strict prompt accuracy, exceeding the strongest standalone LLM baseline, gpt-5.4, by 5.81 percentage points while operating in a lower-cost regime. We then analyze the tokenomics and operational behavior behind this gain, including cost per sample, token composition, useful-output goodput, feedback-loop recovery, latency decomposition, and performance across instruction categories and constraint counts. Our results show that agentic SLM ensembles can trade additional test-time tokens and orchestration overhead for improved instruction-following fidelity, motivating multi-dimensional evaluation protocols for future agentic AI systems.

CCS Concepts

• **Computing methodologies** → **Multi-agent systems**; **Intelligent agents**.

Keywords

Agentic AI, Small Language Model, Model Context Protocol, Tokenomics, Ensemble, IFEval, Beyond Leaderboard Evaluation

1 Introduction

Large language models (LLMs) are increasingly used as components of agentic workflows, where success depends not only on producing plausible answers but also on reliably satisfying explicit constraints, schemas, and protocol requirements [1, 6, 7, 9–11]. This shift exposes a limitation of conventional leaderboard-style evaluation: aggregate accuracy alone does not capture whether a system is practical to deploy when iterative feedback, orchestration, latency, and token cost are part of the inference procedure [4, 5, 8, 14]. Instruction-following benchmarks such as IFEval [16] provide a useful testbed for this setting because they decompose prompts into verifiable constraints involving formatting, length, inclusion/exclusion rules, and multi-instruction composition. At the same time, standalone frontier models can be expensive for repeated agentic inference, while compute-efficient small language models (SLMs) often struggle with complex constraint combinations [3, 15].

We therefore evaluate an agentic ensemble of SLMs implemented using Model Context Protocol (MCP) as the integration framework [2, 12], combining programmatic IFEval checking with an SLM judge-mediated feedback loop. The system first demonstrates that SLM ensembles can be competitive with, and in our experiments outperform, standalone LLM counterparts on instruction-following accuracy. We then shift the analysis beyond leaderboard accuracy to examine the tokenomics and operational behavior that enable this improvement, including cost per sample, relative token composition, useful output goodput, iterative recovery, latency decomposition, category-level performance, and robustness as the number of constraints increases. Our novelty is a multi-dimensional evaluation of agentic SLM ensembles that treats instruction fidelity, token overhead, feedback-loop cost, and deployment efficiency as coupled evaluation targets. The contribution is an empirical case study showing both that SLM ensembles can work well on IFEval and that their benefits cannot be understood from accuracy alone.

2 Ensemble System Design and Evaluation

We developed an agentic SLM ensemble implemented using MCP as the communication layer. The system coordinates multiple compute-efficient SLMs as response generators and uses an SLM judge to support tie-breaking and feedback generation. MCP is used to structure the interaction among generators, judge, and the programmatic IFEval checker, enabling reproducible multi-model execution and iterative correction.

For each IFEval prompt, the ensemble runs for at most three attempts. In the first attempt, all SLM ensemble members independently generate candidate responses to the same multi-constraint prompt. Each candidate is then evaluated using the IFEval programmatic checker. If multiple candidates pass, the SLM judge acts as a stylistic tie-breaker among already compliant outputs. If no candidate satisfies all constraints, the judge converts the violations identified by the programmatic checker into targeted, natural-language corrective feedback for the ensemble members. This process is repeated for up to two corrective attempts.

This design separates three functions that are often conflated in aggregate LLM evaluation: generation, constraint verification, and feedback-guided correction. It also supports a principled Best-of-N interpretation for IFEval: because the benchmark evaluates instruction following rather than domain knowledge, selecting the candidate with maximal constraint satisfaction is an appropriate way to measure the best instruction-compliant output available from the ensemble. We benchmark five ensemble configurations against standalone large and small models from OpenAI, Google, Anthropic, Meta, and Mistral. Full model and ensemble compositions are provided in Table 1.

2.1 Beyond-Leaderboard Evaluation

We evaluate the proposed system on the IFEval benchmark [16], which contains 541 prompts with rule-based instruction checks. To

maintain compatibility with the standard benchmark, we report the primary IFEval accuracy metrics:

- `Strict Prompt-level Accuracy`: the fraction of prompts for which every underlying constraint is satisfied.
- `Strict Instruction-level Accuracy`: the fraction of individual instructions satisfied across the full benchmark.
- `Loose Prompt-level Accuracy`: a prompt-level metric that permits minor formatting variations, such as extra whitespace or markdown artifacts.

Because the goal of this study is to evaluate agentic ensembles as deployable systems rather than only leaderboard entries, we additionally track operational and tokenomic characteristics:

- **Cost per sample**: average API cost required to process one IFEval prompt.
- **Token composition**: the fraction of total tokens assigned to input context, accepted final output (goodput), exploratory sampled outputs, and orchestration/judging.
- **Goodput ratio**: the ratio of useful final-output tokens to total compute tokens consumed by the system.
- **Iteration progress**: cumulative accuracy after each feedback attempt, measuring the marginal utility of the judge-feedback loop.
- **Latency decomposition**: active processing time separated into model inference and orchestration/judging components.
- **Slice-level robustness**: accuracy by IFEval instruction category and by the number of constraints per prompt.

These measurements allow us to ask not only whether an ensemble improves accuracy, but also how much additional compute and orchestration overhead are required to obtain that improvement.

3 Experimental Results

We evaluate our ensembles on the IFEval benchmark. To estimate stochastic variation, each configuration is run five times and reported as mean accuracy with standard deviation ($\mu \pm \sigma$). Mistral and Llama variants were served via the OpenRouter platform [13]. We first establish whether the proposed agentic SLM ensembles are competitive under standard IFEval metrics, and then analyze the tokenomic and operational costs of achieving these gains.

Figure 1 shows that all five ensemble configurations outperform the standalone baselines on strict prompt accuracy. The lowest-performing ensemble, [Ens-5], reaches 95.64%, while the strongest standalone model, gpt-5.4, reaches 91.53%. The best configuration, [Ens-1] with gemini-3.1-flash as judge and gpt-5.4-mini/gemini-3.1-flash as members, achieves 97.34% strict prompt accuracy, 5.81 percentage points above the strongest standalone baseline.

The same pattern holds for instruction and loose prompt metrics, where [Ens-1] reaches 98.3% instruction-level accuracy and 98.4% loose prompt-level accuracy. These complementary metrics, shown in Figure 2, indicate that the gains are not limited to a single strict formatting definition. Together, the standard IFEval metrics establish the viability of the agentic SLM ensemble architecture.

After demonstrating that the proposed SLM ensembles work under standard IFEval metrics, the central evaluation question becomes what operational tradeoffs enable these gains. A leaderboard score alone does not reveal whether the improvement comes from efficient collaboration, repeated sampling, judge overhead, or feedback-loop recovery. We therefore turn to beyond-leaderboard analysis, examining cost, token allocation, useful output goodput, iteration dynamics, latency, and robustness.

Figure 3 shows that the ensembles occupy a favorable region of the cost-performance space: they exceed the accuracy of the strongest standalone LLM while remaining less expensive per prompt than several large-model baselines. Figure 4 shows token allocation breakdown between input, output, exploration, and orchestration overhead. The fact that orchestration tokens (~45–47%) outweigh exploration tokens (~25–31%) proves that the system is not just relying on sampling to find a "lucky" output. Instead, it spends more compute on analyzing and correcting errors than it is on sampling. This strongly supports the agentic self-correction narrative and shows that crossing the 95% accuracy threshold requires investment in orchestration, evaluation and structured feedback generation rather than just letting models blindly generate results in an open loop (zero−shot). Combined with Figure 3, this shows that SLM ensembles can cost less than standalone LLMs despite orchestration overhead, owing to compute-efficient mini-model inference.

The feedback loop also contributes measurable recovery (Figure 5). Across ensembles, cumulative strict accuracy increases over successive attempts, indicating that the judge-feedback mechanism contributes to the recovery of initially invalid outputs. Finally, Figure 6 shows strict prompt accuracy plotted against the ratio of useful output tokens to total generated compute tokens, exposing two completely different clusters with no middle ground. Our ensembles cluster in the top-left quadrant, surpassing the 95% accuracy threshold. Conversely, standalone models cluster heavily on the far right (0.70–0.90 token ratio) and hit a performance ceiling below the 92% mark. This demonstrates that an ensemble of SLMs can outperform a single large model despite token overhead.

4 Conclusions and Future Work

This work evaluates agentic SLM ensembles on IFEval from both a leaderboard and beyond-leaderboard perspective. The best ensemble achieves 97.34% strict prompt accuracy, exceeding the strongest standalone LLM baseline by 5.81 percentage points. This demonstrates that orchestrated SLMs, when coupled with programmatic IFEval checking and SLM-judge-mediated feedback, can outperform standalone LLMs on instruction-following tasks. More importantly, our tokenomic analysis shows that this improvement is enabled by tradeoffs: orchestration/judging overhead, iterative recovery, and latency costs. These results support the need for multi-dimensional evaluation of agentic systems, where accuracy, cost, token goodput, latency, and robustness are evaluated jointly. Future work will extend the framework to richer consensus mechanisms, including multi-agent panel debates and LLM-as-a-Jury methods, and to larger model pools. We also plan to study science-oriented agentic workflows, where reliable constraint following may extend beyond schema compliance to domain-specific validity checks.

Acknowledgments

Research presented in this report was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20250637DI, 20250638DI, and 20250639DI.

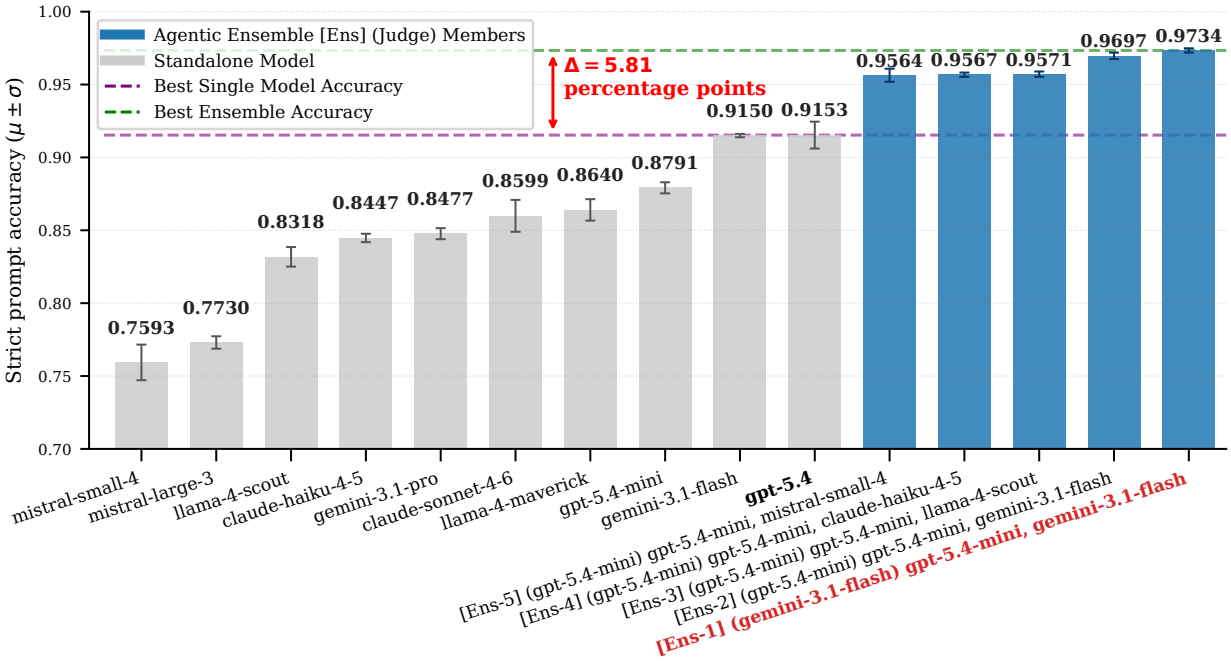


Figure 1: Strict prompt accuracy ($\mu \pm \sigma$) on the IFEval benchmark ($N = 541$). For ensembles, identifiers denote the judge model in parentheses followed by the ensemble members. Horizontal dashed lines and bold labels denote the performance winners in each category, highlighting a gain of 5.81 percentage points for the proposed ensemble model over the strongest single-model baseline.

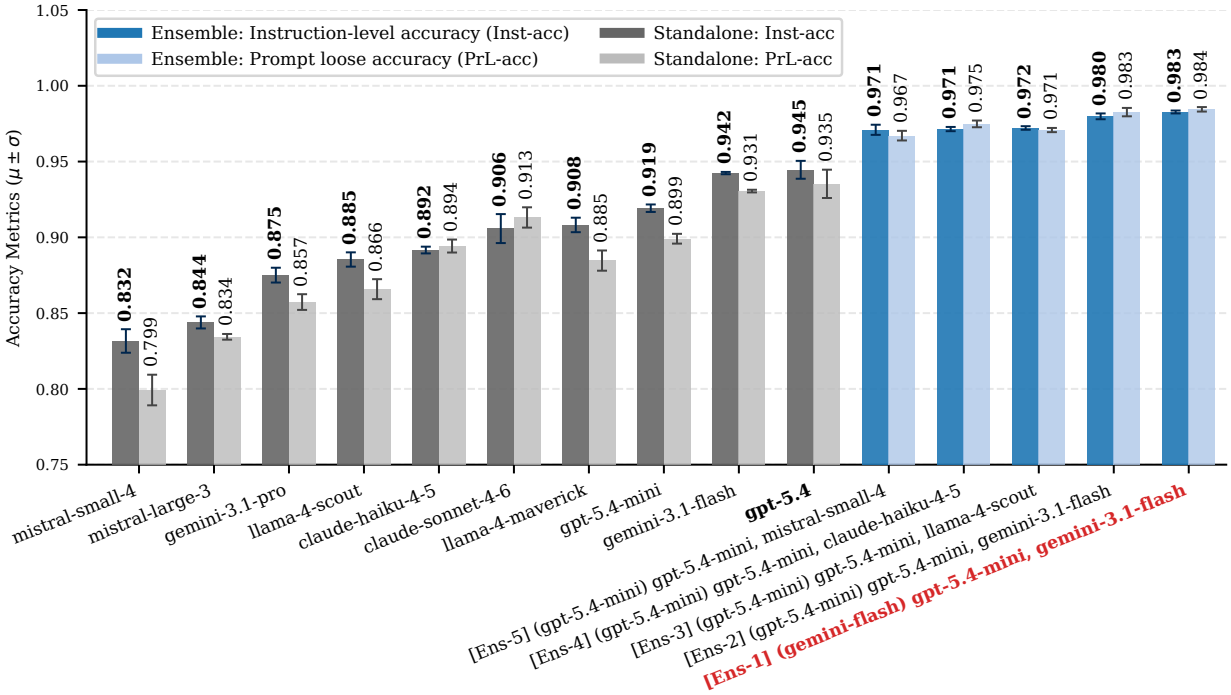


Figure 2: Instruction-level strict accuracy (Inst-acc) and prompt-level loose accuracy (PrL-acc) across standalone models and ensembles ($\mu \pm \sigma$) on the IFEval benchmark ($N = 541$). Ensembles achieve higher overall accuracy and a significantly narrower gap between accuracy metrics, indicating ensemble robustness.

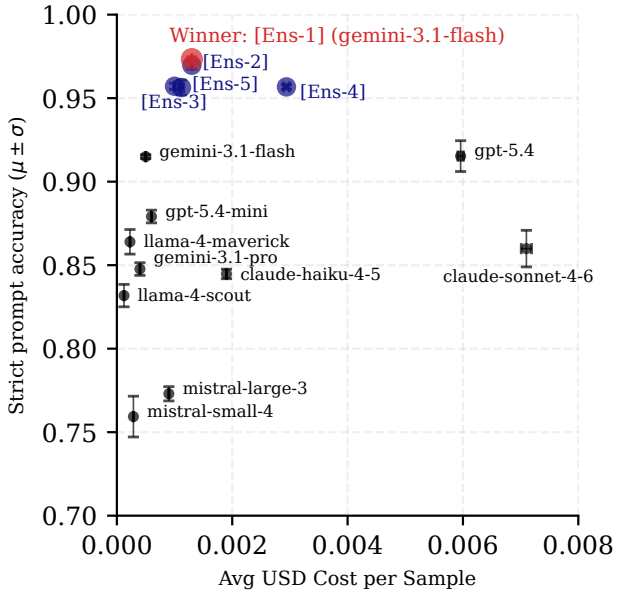


Figure 3: Strict prompt accuracy vs. average cost per sample ($\mu \pm \sigma$). The winning ensemble [Ens-1] (red) with gemini-3.1-flash as a judge achieves the highest accuracy while maintaining a cost advantage over large models like gpt-5.4. The ensemble indices ([Ens-n]) match the ensembles in Figure 1.

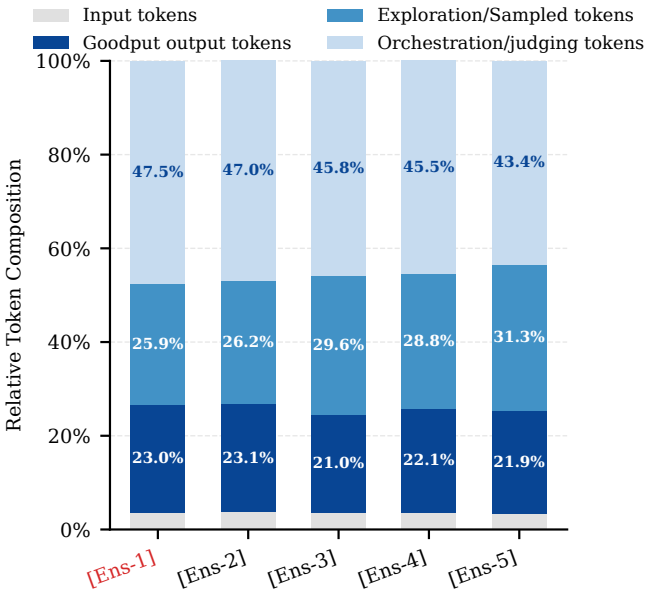


Figure 4: Relative token composition per processing sample/prompt. The breakdown illustrates the average allocation of tokens across different ensembles. The ensemble indices ([Ens-n]) match the ensembles in Figure 1.

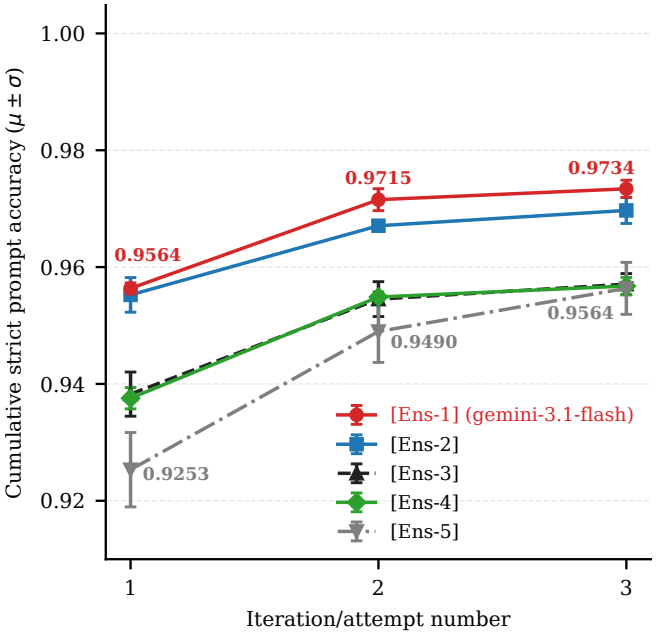


Figure 5: The cumulative prompt-level strict accuracy progress over iterations. The upward trajectory demonstrates the system's capacity to identify instruction violations and provide corrective feedback to ensemble members, recovering valid outputs. The ensemble indices ([Ens-n]) match the ensembles in Figure 1.

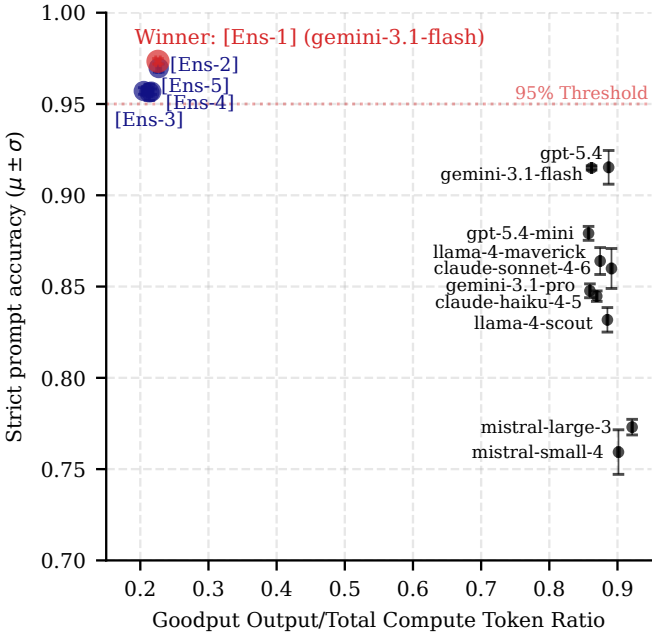


Figure 6: Strict prompt accuracy vs. Goodput Output/Total Compute Token Ratio. The plot highlights ensembles clustering in the top-left quadrant, surpassing the 95% accuracy threshold. This demonstrates that the ensemble of smaller models can achieve higher accuracy than a single large model, even though the ensemble's token overhead is higher.

References

[1] Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan Alrashed, Shaykhah Alsubaie, Yousef Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, Areeb Alowisheq, et al. 2024. When Benchmarks are Targets: Revealing the Sensitivity of Large Language Model Leaderboards. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (ACL '24)*. Association for Computational Linguistics, Bangkok, Thailand, 13787–13805. doi:10.18653/v1/2024.acl-long.744

[2] Anthropic. 2024. Introducing the Model Context Protocol. https://www.anthropic.com/news/model-context-protocol. Accessed: 2026-05-18.

[3] Peter Belcak, Greg Heinrich, Shizhe Diao, Yonggan Fu, Xin Dong, Saurav Muralidharan, Yingyan Celine Lin, and Pavlo Molchanov. 2025. Small Language Models are the Future of Agentic AI. *arXiv preprint arXiv:2506.02153* (2025). arXiv:2506.02153 [cs.CL]

[4] Lingjiao Chen, Matei Zaharia, and James Zou. 2023. FrugalGPT: How to Use Large Language Models While Reducing Cost and Improving Performance. *arXiv preprint arXiv:2305.05176* (2023). arXiv:2305.05176 [cs.CL]

[5] Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot Arena: An Open Platform for Evaluating LLMs by Human Preference. *arXiv preprint arXiv:2403.04132* (2024). arXiv:2403.04132 [cs.CL]

[6] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770* (2023). arXiv:2310.06770 [cs.CL]

[7] Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Chong Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Ruslan Salakhutdinov, and Daniel Fried. 2024. VisualWebArena: Evaluating Multimodal Agents on Realistic Visual Web Tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

[8] Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, Benjamin Newman, Binhang Yuan, Bobby Yan, Ce Zhang, Christian Cosgrove, Christopher D. Manning, Christopher Ré, Diana Acosta-Navas, Drew A. Hudson, Eric Zelikman, Esin Durmus, Faisal Ladhak, Frieda Rong, Hongyu Ren, Huaxiu Yao, Jue Wang, Sang Michael Xie, Shibani Santurkar, Surya Ganguli, Tatsunori Hashimoto, Thomas Icard, Tianyi Zhang, Vishrav Chaudhary, William Wang, Xuechen Li, Yifan Mai, Yuhui Zhang, and Yuta Koreeda. 2023. Holistic Evaluation of Language Models. *Transactions on Machine Learning Research* (2023).

[9] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2023. AgentBench: Evaluating LLMs as Agents. *arXiv preprint arXiv:2308.03688* (2023). arXiv:2308.03688 [cs.CL]

[10] Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2023. GAIA: A Benchmark for General AI Assistants. *arXiv preprint arXiv:2311.12983* (2023). arXiv:2311.12983 [cs.AI]

[11] Justin K. Miller and Wenjia Tang. 2025. Evaluating LLM Metrics through Real-World Capabilities. *arXiv preprint arXiv:2505.08253* (2025). arXiv:2505.08253 [cs.CL]

[12] Model Context Protocol. 2025. Model Context Protocol Specification. https://modelcontextprotocol.io/specification/2025-06-18. Accessed: 2026-05-18.

[13] OpenRouter. 2026. *OpenRouter: A Unified API for Large Language Models*. Retrieved May 18, 2026 from https://openrouter.ai

[14] Saeid Sheikhi, Lauri Lovén, and Panos Kostakos. 2026. Beyond the Leaderboard: A Survey of the Science of Evaluation, Benchmarking, and Methodologies for Large Language Models. *IEEE Access* (2026). doi:10.1109/ACCESS.2026.3686088

[15] Shreyas Subramanian, Vikram Elango, and Mecit Gungor. 2025. Small Language Models (SLMs) Can Still Pack a Punch: A Survey. *arXiv preprint arXiv:2501.05465* (2025). arXiv:2501.05465 [cs.CL]

[16] 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. *arXiv preprint arXiv:2311.07911* (2023). arXiv:2311.07911 [cs.CL]

A Appendix

A.1 Model and Ensemble Configurations

Model and ensemble configurations are shown in Table 1.

A.2 Category-Level and Constraint-Complexity Analysis

Figure 7 reports strict prompt accuracy across IFEval general instruction categories. The results show that ensemble gains are not uniform across instruction types: the largest benefits occur in categories where failures are often recoverable through feedback, such as formatting, content-detection, and multi-condition constraints. This supports the interpretation that the feedback loop improves instruction fidelity by correcting identifiable constraint violations rather than merely increasing generic model capability.

Figure 8 further evaluates performance as a function of the number of instructions per prompt. As the number of simultaneous constraints increases, standalone model accuracy decreases more sharply, whereas the agentic ensembles retain higher strict accuracy. This suggests that ensemble feedback is particularly useful in compositional instruction-following settings, where a response may partially satisfy the prompt but fail one or more atomic constraints.

A.3 Latency Decomposition

Figure 9 decomposes average active processing time into inference and orchestration/judging components under sequential execution (max_concurrency = 1). This provides a conservative latency estimate because ensemble members and feedback cycles are not executed concurrently, although the implementation supports higher-concurrency runs. Even in this sequential setting, the leading SLM ensembles remain practical: they introduce judge and feedback overhead, but still achieve lower active processing time than the strongest standalone LLM baseline, gpt-5.4, while improving strict instruction-following accuracy. This highlights a key tokenomics tradeoff: additional coordination tokens and processing steps can be worthwhile when they are spent on compute-efficient component models.

A.4 The Non-ASCII Roulette Anomaly

During our systematic analysis of the underlying IFEval testing suite, we identified a small source of stochasticity in the benchmark's native evaluation routine, which we term the "Non-ASCII Roulette" anomaly. Specifically, the evaluation script applies a character replacement preprocessing step in which selected non-ASCII characters, as well as certain other characters whose code points fall outside predefined ranges, may be randomly replaced with standard ASCII characters. Importantly, this preprocessing is applied to the instruction constraints, not to the model output. For example, in a character-counting instruction such as "the character '#' should appear at least 3 times," the specified target character may be randomly replaced before compliance is checked.

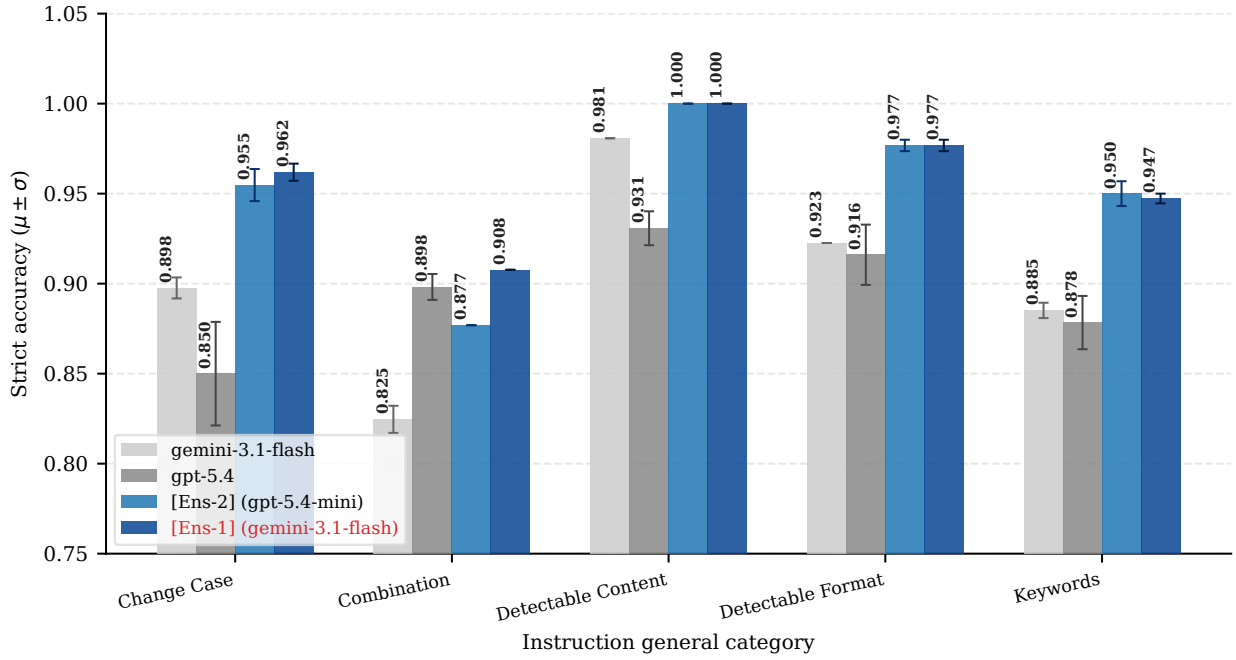
As a result, the checker remains rule-based in its pass/fail logic, but the instruction preprocessing step can introduce randomness into what is being checked for affected character-based instructions. Consequently, repeated evaluations of the same model output may produce slightly different evaluation scores when the corresponding instruction contains affected characters.

To preserve comparability with existing IFEval leaderboards and prior reported results, we maintained the original evaluation parameters unchanged. However, this implementation detail should be considered when interpreting small differences in instruction-

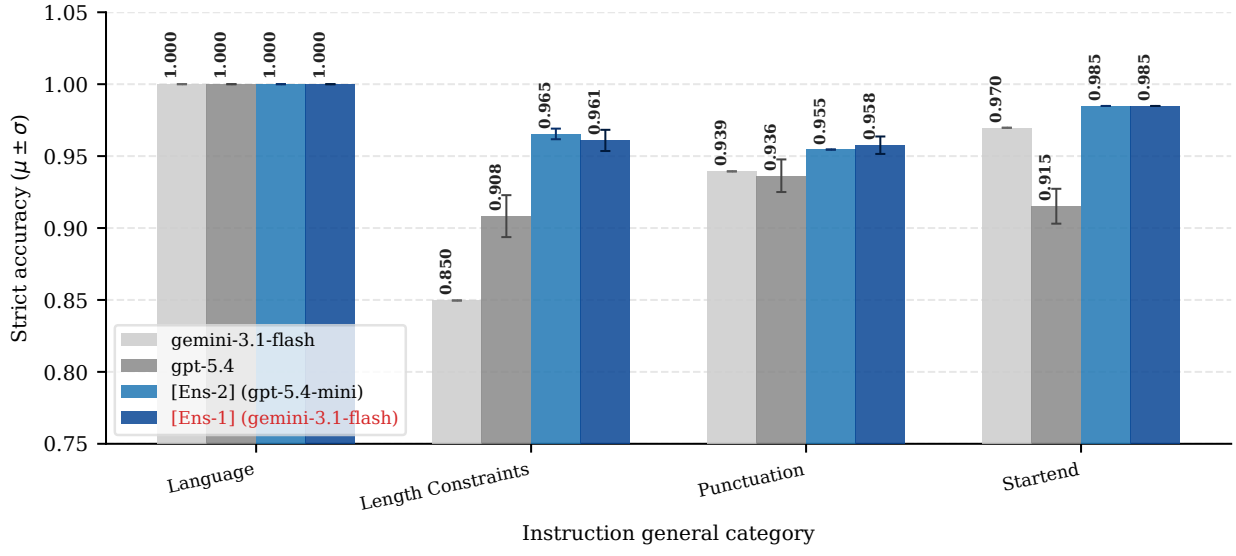
Table 1: Evaluated large and small language models (LLMs and SLMs) and ensembles of SLMs.

Large Language Model (LLM)	Small Language Model (SLM)	
gpt-5.4	gpt-5.4-mini	
gemini-3.1-pro-preview	gemini-3.1-flash-lite-preview	
claude-sonnet-4-6	claude-haiku-4-5	
llama-4-maverick	llama-4-scout	
mistral-large-3	mistral-small-4	
Ensemble composition: [Ens-n] (Judge model) ensemble member names		
[Ens-1]	(gemini-3.1-flash-lite-preview)	gpt-5.4-mini, gemini-3.1-flash-lite-preview
[Ens-2]	(gpt-5.4-mini)	gpt-5.4-mini, gemini-3.1-flash-lite-preview
[Ens-3]	(gpt-5.4-mini)	gpt-5.4-mini, llama-4-scout
[Ens-4]	(gpt-5.4-mini)	gpt-5.4-mini, claude-haiku-4-5
[Ens-5]	(gpt-5.4-mini)	gpt-5.4-mini, mistral-small-4

and prompt-level metrics, particularly for instructions involving affected non-ASCII or special characters.



(a)



(b)

Figure 7: Accuracy ($\mu \pm \sigma$) breakdown by IFEval instruction general categories. The breakdown is partitioned into (a) and (b). We show the top two standalone baselinesand two leading ensembles. The ensemble indices ([Ens-n]) match the ensembles in Figure 1.

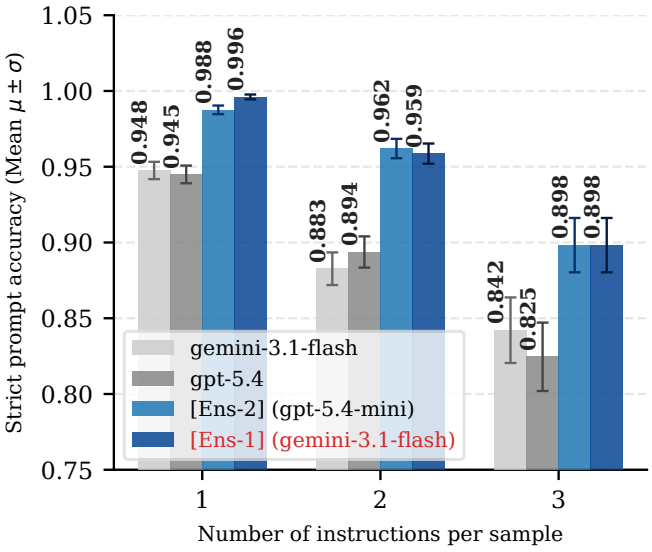


Figure 8: Strict prompt accuracy vs. sample complexity (defined as the number of instructions per prompt). We compare the top two standalone baselines with the leading ensembles. The results show that ensembles maintain higher accuracy at maximum complexity compared to standalone models. The ensemble indices ([Ens-n]) match the ensembles in Figure 1.

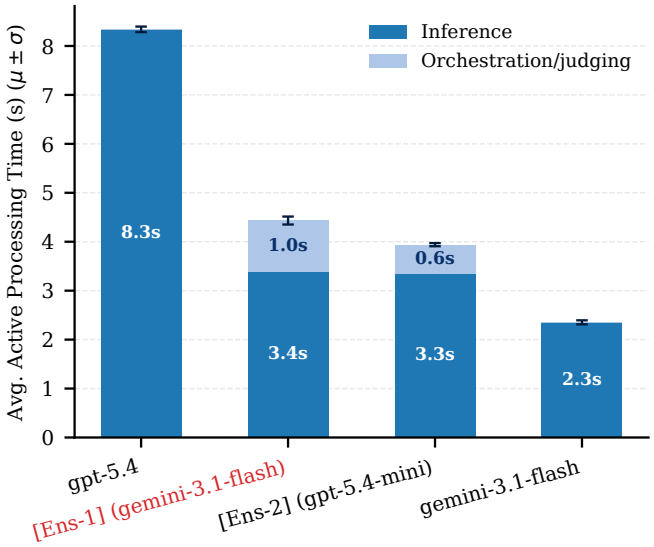


Figure 9: Average active processing time breakdown across the top two standalone models and the leading ensembles ($\mu \pm \sigma$) evaluated under sequential execution (max_concurrency = 1). Total latency is partitioned into inference and orchestration/judging cycles.