# Adaptive Self-Consistency for Long-Form Factuality on Long-Tail Facts

**Daniil Moskovskiy[1,2], Rafael Grigoryan[3], Evgenii Shuranov[4], Wenshuai Yin[5], and Alexander Panchenko[2,1]**
[1]AIRI, [2]Skoltech, [3]MSU, [4]ITMO University, [5]HSE
A.Panchenko@skol.tech

## Abstract

Long-form question answering for rare entities remains challenging for large language models, while inference-time methods for improving factual precision in this setting are underexplored. We study multi-sample self-consistency for long-form generation across various Wikipedia entity popularity tiers using the RiDiC dataset (Braslavski et al., 2026), using fact-level evaluation. We compare several aggregation strategies and find that LLM-based aggregation consistently outperforms voting and selection baselines, including a compute-matched best-of-K logprob selector, across languages and domains, with most gains achieved at relatively small sample budgets. Motivated by this early-saturation effect, we introduce an adaptive policy that predicts an appropriate generation budget from pre-generation features. Against a conservative full-budget self-consistency reference (99 generations per question), the adaptive policy retains over 99% of factual precision while reducing the number of sampled generations by 2–29% across various language–domain pairs. We further investigate how these savings depend on the reference budget and identify the regime where adaptive routing helps most.

## 1 Introduction

Large language models (LLMs) are increasingly used in information-seeking settings that require long, coherent answers rather than short factual snippets. In such settings, factuality failures are especially harmful: long responses may remain fluent and persuasive while still containing multiple unsupported or hallucinated statements (Maynez et al., 2020; Ji et al., 2023). Long-form factuality is now recognized as a distinct evaluation challenge, since correctness must be assessed over many fine-grained claims rather than a single final answer (Min et al., 2023; Zhao et al., 2024; Wei et al., 2024). This challenge becomes more severe for unpopular entities, where factual recall is often weaker and purely parametric generation tends to be less reliable (Mallen et al., 2023). We operationalize popularity using RiDiC's Wikipedia-pageview tiers: "tail" denotes low-pageview entities, while head and torso provide higher-popularity comparison groups (Braslavski et al., 2026).

Self-consistency offers one inference-time remedy: rather than relying on a single sampled response, generate multiple candidate completions and select or synthesize a final answer from them (Wang et al., 2023; Chen et al., 2023; Thirukovalluru et al., 2024). In practice, the saturation point is generally not known in advance since it depends on the task and the model one uses. A conservative large-budget regime is wasteful when most of the gain is realized early. Little is yet known about how to cut this waste for long-form factual generation.

Studying this setting requires an evaluation protocol that combines long-form generation, fact-level verification, multilingual coverage, domain diversity, and explicit Wikipedia-pageview popularity tiers. Existing resources satisfy these requirements only partially. Benchmarks with popularity annotations, such as PopQA, are centered on short-answer question answering rather than long-form generation (Mallen et al., 2023). Long-form factuality benchmarks better match the generation setting (Min et al., 2023; Zhao et al., 2024), but they often do not provide controlled popularity strata or domain splits, making it harder to isolate long-tail effects (Zhao et al., 2024; Wei et al., 2024; Vazhentsev et al., 2026). We therefore evaluate on the existing RiDiC benchmark, whose entity-centric prompts, multilingual coverage, domain splits, and Wikipedia-pageview tiers support this controlled analysis (Braslavski et al., 2026).

We ask three questions: (i) whether multi-sample self-consistency improves long-form fac-

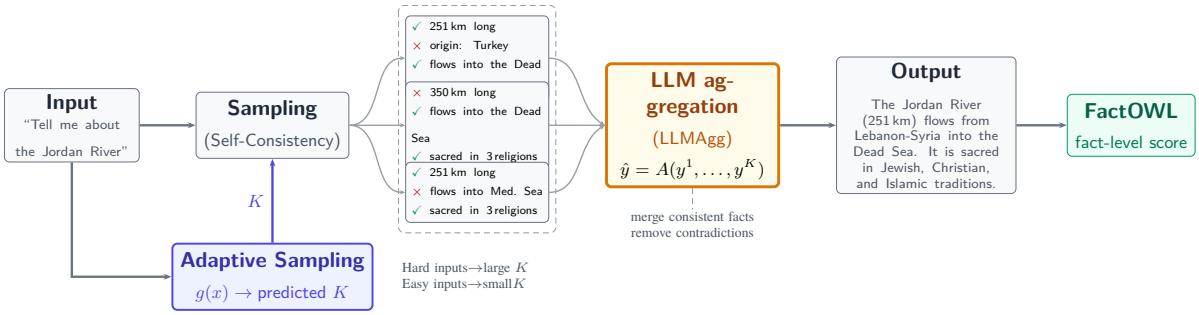


Figure 1: Overview of the proposed pipeline. An input prompt is sampled multiple times via self-consistency. An adaptive controller predicts the generation budget K from the input features. Candidate generations are aggregated via LLMAgg, and the final output is assessed with fact-level evaluation.

tual precision across these popularity tiers, (ii) which aggregation strategy best consolidates multiple long-form generations, and (iii) how much of the factuality gain survives when the sampling budget is allocated adaptively rather than fixed at the full-budget reference.

Our contributions are empirical rather than a new benchmark or factuality evaluator: (i) we compare content-level LLM aggregation with selection and voting baselines for long-form factuality on RiDiC; (ii) we characterize budget–factuality curves across languages, domains, and popularity tiers; and (iii) we evaluate a lightweight pre-generation policy that routes inputs to a sampling budget relative to a conservative full-budget reference.

Within this single-generator setting, LLMAgg substantially outperforms single generation, voting baselines, and a compute-matched best-of-K logprob selector across domains and languages. The budget-scaling curves saturate early but vary across domains and languages. We therefore add a simple routing policy that predicts the generation budget before candidate generation. Against the full-budget reference, the policy retains nearly all factual precision while reducing the number of sampled generations.

## 2 Related Work

**Self-Consistency** improves language-model reliability by sampling multiple outputs and aggregating them (Wang et al., 2023). Most prior work studies this idea in reasoning and short-answer settings, where aggregation can often be reduced to voting over final answers. Universal Self-Consistency extends the paradigm to open-ended generation through response selection (Chen et al., 2023), while Atomic Self-Consistency aggregates selected atomic subparts of candidate generations (Thirukovalluru et al., 2024).

Our setting differs from these scenarios. Long-form generations contain many partially correct and partially incorrect statements, so agreement at the response level is a weak proxy for factual correctness. Aggregation strategies can also behave very differently when candidates overlap without being identical. We therefore study aggregation directly, comparing LLM-based aggregation against selection and voting baselines under fact-level evaluation.

**Long-Form Factuality Evaluation** Long-form factuality is now treated as a distinct evaluation problem, because generated responses often contain many atomic claims whose correctness must be assessed individually rather than through exact-match style metrics (Maynez et al., 2020; Min et al., 2023; Ji et al., 2023). FActScore formalizes this perspective by decomposing long-form outputs into atomic facts and measuring the fraction that can be supported (Min et al., 2023). Other recent benchmarks, such as WildHallucinations (Zhao et al., 2024), further emphasize the difficulty of maintaining factual consistency in entity-centric long-form generation.

These benchmarks establish the need for fact-level evaluation, but they only partially match the setting we study here. Popularity-aware resources such as PopQA target short-answer factual recall rather than long-form generation (Mallen et al., 2023), while long-form factuality benchmarks do not always provide controlled popularity partitions together with multilingual and domain-structured evaluation. Our work builds on this literature by studying inference-time factuality across pageview-defined popularity tiers.

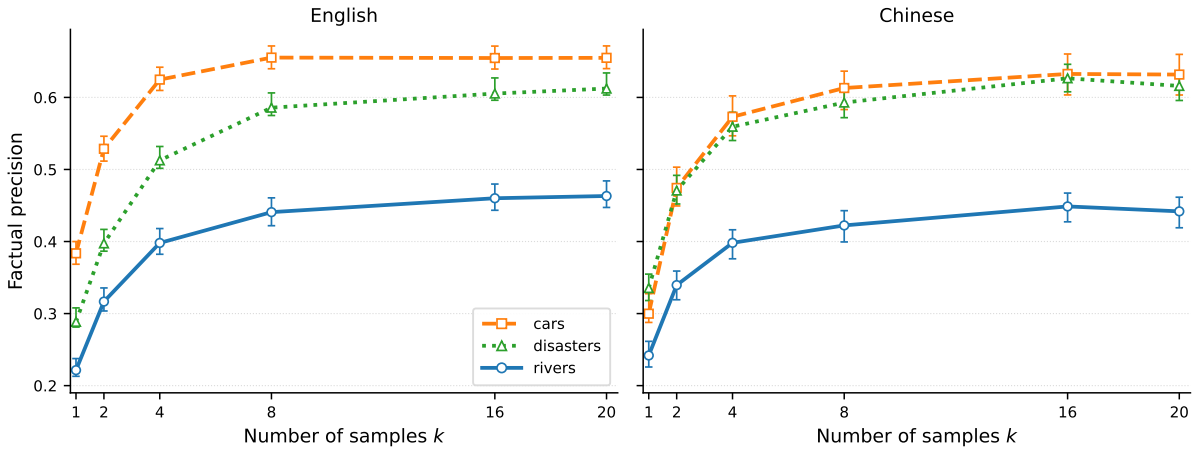


Figure 2: LLMAgg budget-scaling curves across sample budgets. Performance improves sharply at small K and saturates by roughly K = 8–16 in both languages, with only marginal changes beyond K = 20. The saturation point is heterogeneous across domains, which motivates input-conditioned routing rather than a single fixed budget.

**Adaptive Inference and Test-Time Scaling** Recent work shows that additional inference-time computation can improve model outputs. Test-time scaling results suggest that increasing the number of samples can substantially improve performance, in some cases more effectively than scaling model size alone (Snell et al., 2024; Wu et al., 2025). These gains, however, are heterogeneous across inputs. Adaptive methods respond by allocating more compute only where it is expected to help.

This idea was explored in self-consistency and confidence-aware inference, including early-stopping and reliability-aware variants of multi-sample decoding (Taubenfeld et al., 2025; Kim et al., 2026; Marina et al., 2026). We take the same perspective but target long-form factual generation evaluated at the level of atomic claims. In this setting, the main question is not only whether more samples help, but how much of the aggregation gain can be preserved when the sampling budget is routed adaptively relative to a conservative full-budget reference.

## 3 Method

We combine self-consistency, LLM-based aggregation, and adaptive inference into a compute-aware framework for long-form factuality. Given an input prompt $x$, we generate multiple candidate responses, aggregate them into a single output, and evaluate factuality at the level of individual facts. Figure 1 illustrates the overall pipeline.

**Self-Consistency Sampling** Given an input prompt $x$, we sample K independent responses:

$$y^{(1)}, y^{(2)}, \ldots, y^{(\mathrm{K})} \sim p_\theta(y \mid x).$$

This diversity is what lets us identify factual information that is consistent across samples. Unlike standard single-generation inference, self-consistency allows the model to explore multiple plausible answers, some of which may correct errors present in others.

**LLM-based Aggregation** To obtain a final response, we aggregate the sampled outputs using an LLM-based aggregation function (LLMAgg):

$$\hat{y} = \mathcal{A}(y^{(1)}, \ldots, y^{(\mathrm{K})}).$$

The aggregation model is prompted to merge consistent factual statements across samples while removing contradictions and unsupported claims. This step leverages redundancy in the sampled outputs to improve factual precision.

As shown in Figure 1, aggregation operates on multiple sampled responses. Unlike majority voting or heuristic selection, LLMAgg works on the text itself, so it can merge complementary facts from different generations.

**Adaptive Sampling** Instead of using a fixed full-budget number of samples $\mathrm{K}_{\text{full}}$, we employ an adaptive strategy that predicts a generation budget for each input: $\mathrm{K} = g(x)$, where $g$ is a lightweight classifier designed to approximate the marginal benefit of additional sampling.

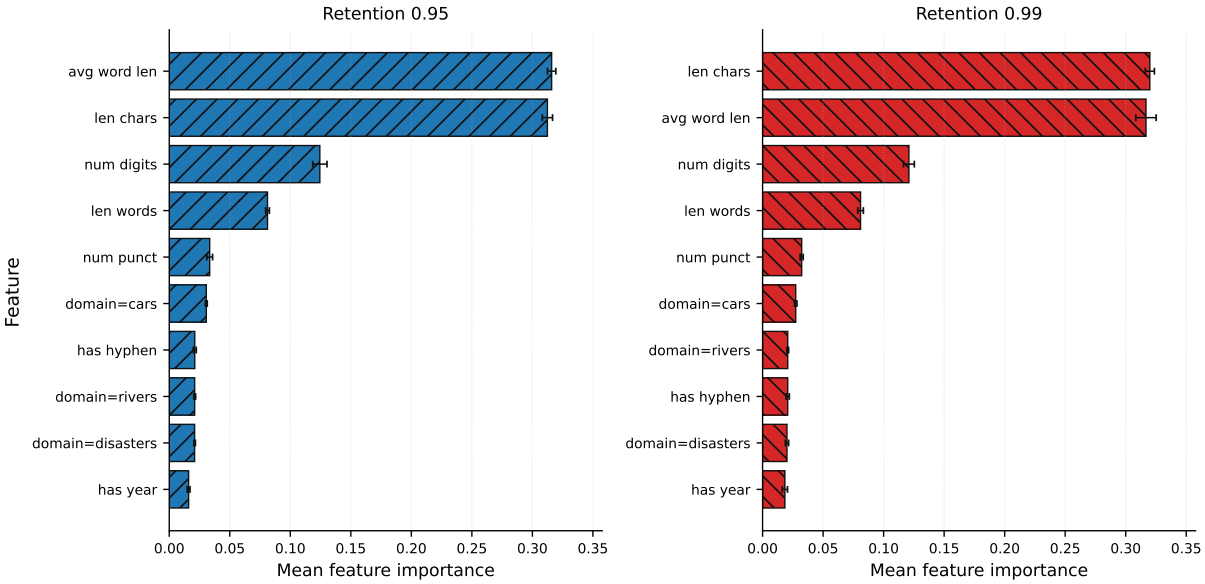


Figure 3: Feature importance for the adaptive controller. Input length and lexical-complexity features dominate, while language and domain indicators contribute relatively little. Error bars show the standard deviation of feature importance across runs.

Different inputs have different levels of factual uncertainty. For difficult inputs, more samples increase the chance of observing correct facts, while for easier inputs, fewer samples are sufficient. Because the saturation point is heterogeneous across domains and not known before evaluation, an input-conditioned policy can reduce reliance on a single conservative budget within the evaluated distribution.

The result is a cost–factuality trade-off that spends extra samples only where they are needed.

## 4 Experimental Setup

**Dataset** Evaluating long-form factuality across pageview-defined popularity tiers requires more than a standard entity benchmark. Popularity-aware resources such as PopQA provide valuable supervision for knowledge frequency, but they target short-answer question answering rather than long-form generation (Mallen et al., 2023). Long-form factuality benchmarks are closer to our setting, but they typically do not provide explicit domain splits together with controlled popularity partitions, which limits their usefulness for analyzing long-tail behavior in a structured way (Zhao et al., 2024).

We use RiDiC (Braslavski et al., 2026), an existing benchmark for entity-centric long-form factuality evaluation with explicit popularity stratification. RiDiC covers three domains (rivers, cars, and disasters) and two languages (English and Chinese). The nominal test split contains 1000 examples per domain. We use its public test metadata and the corresponding public self-consistency exports and their Chinese counterparts.[1,2,3] After filtering for usable entity titles and Wikipedia contexts, the effective counts are $1000/997/1000$ for English rivers/cars/disasters and $723/400/723$ for the corresponding Chinese cells. Table 1 presents an example from the RiDiC benchmark, illustrating the variability and factual inconsistencies in model outputs.

Thus, `test split` refers to the public test partition; reported FactOWL scores use the usable-context subset of that partition. Its `head`, `torso`, and `tail` tiers are derived from the `popularity_part_sector`, which is based on Wikipedia page views. We use this stratification as an analysis axis: it reveals whether factuality and budget behavior persist under the benchmark's controlled variation in entity popularity.

**Generation** We use Llama 3.1 8B Instruct (Team, 2024) with stochastic decoding to induce diversity across generations. The same generator is used for English and Chi-

[1] hf.co/datasets/s-nlp/RiDiC_rivers_self_consistency_99_generations

[2] hf.co/datasets/s-nlp/RiDiC_cars_self_consistency_99_generations

[3] hf.co/datasets/s-nlp/RiDiC_bad_weather_self_consistency_99_generations

nese to hold model capacity fixed across the language comparison; we do not claim language-equivalent performance. The saved exports were generated before this study, and their original temperature/top-$p$ settings are not recorded in the available artifacts; consequently, we do not claim to reproduce those decoding parameters here. For the scoring and aggregation reruns, we use vLLM in bfloat16 with tensor parallelism set to 1, GPU memory utilization 0.8, model seed 0, and the maximum of 256 generated tokens. LLMAgg is a separate call to the same Llama checkpoint, with temperature 0 and maximum 256 new tokens; it is not the generator's original completion call. In the full self-consistency setting, we use the available 99 candidate completions per input and set $K_{full} = 99$ as a conservative full-budget reference. Budget-scaling curves use nested prefixes, meaning the first K candidates from this fixed ordered pool, for $K \in \{1, 2, 4, 8, 16, 20\}$; no new sampling is performed at each curve point. Decoding and prompt settings are held fixed across rerun methods.

**Compared Methods** We report six conditions in Table 2. **Single** uses one generated completion and performs no aggregation. **Best samp.** is a non-deployable diagnostic that selects the factuality-best individual candidate after scoring. **MC** (majority co-occurrence) is a response-level agreement baseline: it votes on whether a candidate is supported rather than decomposing responses into facts. **MF** performs majority fact-level voting over normalized FactOWL-derived atomic facts; an atom is retained when it occurs in at least $\lceil \frac{K}{2} \rceil$ candidate responses, and its saved support decision is the majority of its FactOWL decisions. **SP** (SeqProb) is a selector that scores each candidate by its mean token log-probability under the same Llama generator and chooses the highest-scoring candidate. **LLMAgg** uses a separate, deterministic-temperature Llama call to synthesize a final response from multiple candidates. Unlike selection-based methods, it can combine information across generations rather than choosing a single candidate. The compute-matched **Best-of-K logprob** selector chooses the candidate with the highest average token logprob from the same 20-sample pool as LLMAgg@K $=$20. We report that matched K $=$20 comparison in Appendix B, rather than Table 2, which reports the full-budget LLMAgg@K $=$99 comparison.

**Evaluation** We use FactOWL[4] (Sakhovskiy et al., 2026) as a fact-level evaluation framework that decomposes long-form outputs into atomic claims and estimates factual precision from their support status. In our configuration, the Llama 3.1 8B Instruct model performs both atomic-fact generation and fact verification, with temperature 0, a 256−token generation limit for atomic-fact extraction, and an 8−token verification output limit. Using the same model as both extractor and verifier follows a broader practice of treating LLMs as annotators of textual properties, from toxicity labels to factual support (Moskovskiy et al., 2024, 2025). Verification uses the entity's supplied Wikipedia page as context through the `retrieval+llama` configuration; the pipeline does not query an alternative evidence corpus for these scores. Each extracted fact is mapped to a binary support decision, and the response score is the mean support indicator over its extracted facts.

We report factual precision as the main metric. Coverage and retention curves are computed from per-topic scoring outputs. In the main results, **Single** is an independently sampled one-generation baseline, whereas $K = 1$ denotes the first nested prefix of the self-consistency candidate pool; they are therefore not expected to match exactly. Because **Best samp.** is restricted to individual sampled candidates whereas LLMAgg writes a new response from the pool, it is a diagnostic rather than an upper bound on LLMAgg. This comparison measures factual precision, not recall: by combining supported claims from different candidates and omitting unsupported ones, LLMAgg can produce a response with a higher fraction of supported claims than any individual candidate.

**Uncertainty Estimation** We estimate uncertainty with nonparametric bootstrap confidence intervals for the main metrics. For each setting, we resample test instances with replacement 1000 times and compute percentile 95% confidence intervals for factual precision, sampled-generation savings, and score retention. We do not perform a separate pairwise significance test; the intervals are descriptive uncertainty estimates and are not interpreted as formal hypothesis-test results.

**Adaptive Policy** To reduce sampled-generation cost relative to the full-budget reference, we train a random-forest classifier that predicts an appro-

[4] https://github.com/s-nlp/factowl

| **Entity:** Xiaomi SU7 | |
|---|---|
| **Model Output A (hallucinated):** The Xiaomi SU7 is a subcompact SUV based on the SAIC MG ZS platform, featuring a 1.5-liter engine and manual transmission.<br>**Model Output B (partially correct):** The Xiaomi SU7 is an electric SUV launched in 2022, offering a range of up to 600 km and competing with vehicles such as the Tesla Model Y.<br>**Model Output C (more accurate):** The Xiaomi SU7 is an all-electric sedan unveiled in 2023, built on Xiaomi's proprietary platform, with high-performance variants exceeding 600 horsepower and long driving range. | |
| **Example Facts:** | |
| • The Xiaomi SU7 is an electric vehicle. | (True) |
| • It is a subcompact SUV based on MG ZS. | (False) |
| • It was unveiled in 2023. | (True) |
| • It has a 1.5L combustion engine. | (False) |
| • It competes with Tesla Model Y. | (Partially True) |

Table 1: Example from RiDiC showing inconsistent generations for the same entity.

priate generation budget from the allowed curve values. The classifier uses 300 trees, minimum leaf size 3, balanced-subsample class weighting, and seed 13. Features include prompt character/word/token lengths, average word length, digit and punctuation counts, year and question-marker indicators, language and domain indicators, and a domain-level budget prior derived from the budget-scaling curves. Full feature definitions are given in Appendix E. We use five stratified shuffle splits with a 30% test fraction for the multilingual policy; each fold uses the same split seed 13 and model seed 13 plus the fold index. The policy is pre-generation: it uses no partial output, model-confidence, or inter-candidate-disagreement feature. Feature importance is dominated by input length (about 0.31) and average word length (about 0.32), while domain and language features contribute less than 0.03. This indicates that the policy relies primarily on input complexity rather than domain identity.

We intentionally use a simple model to isolate whether pre-generation signals alone are sufficient to capture variation in required compute relative to the full-budget reference.

We report two operating points, corresponding to retention targets of 0.95 and 0.99. These are interpretable practical tolerances: allowing at most 5% or 1% loss relative to the reference. They are operating points rather than statistically optimized thresholds. We define `compute_savings` as the relative reduction in sampled generations with respect to the full-budget LLMAgg reference at $K_{\text{full}} = 99$, `score_retention` as $\text{Fact}(\hat{y}_{\text{adaptive}})/\text{Fact}(\hat{y}_{\text{full}})$, and `coverage` as the fraction of examples meeting the retention target.

## 5 Results

Aggregation drives factuality gains: LLMAgg is the strongest deployable condition in every language–domain cell of Table 2. The selection-based baselines make this concrete. For English, SeqProb improves over Single but remains well below LLMAgg. At the $K = 20$ setting, which we report in Appendix B, the compute-matched best-of-K logprob selector also gains over Single in every cell but still trails LLMAgg by 0.14–0.29 factual-precision points. Majority-style aggregation over completions or fact decisions does no better. In all cases the gains come from consolidating partially correct material spread across several candidates, rather than from picking a single one.

The same ordering among deployable conditions recurs across both languages and all three domains despite substantial differences in absolute difficulty. In English, LLMAgg can exceed the best-sampled diagnostic because the latter is restricted to an existing response, whereas LLMAgg synthesizes a new response from the candidate pool. This is not a recall effect or a violated oracle bound: FactOWL scores the factual precision of the newly written response. As Figure 1 illustrates, synthesis can combine supported claims from several candidates while omitting unsupported ones, yielding a higher fraction of supported claims than any individual response.

The best-sampled diagnostic is higher in several Chinese cells. The largest gap is ZH rivers, where

| Lang | Domain | Single | Best samp. | MC | MF | SP | LLMAgg |
|---|---|---|---|---|---|---|---|
| EN | rivers | 0.211 ± 0.012 | 0.450 ± 0.015 | 0.053 ± 0.014 | 0.088 ± 0.008 | 0.233 | **0.462** ± 0.019 |
| | cars | 0.344 ± 0.014 | 0.622 ± 0.013 | 0.218 ± 0.026 | 0.146 ± 0.012 | 0.412 | **0.653** ± 0.016 |
| | disasters | 0.283 ± 0.012 | 0.574 ± 0.012 | 0.112 ± 0.020 | 0.111 ± 0.010 | 0.352 | **0.620** ± 0.015 |
| **EN** | macro avg. | 0.279 | 0.549 | 0.128 | 0.115 | 0.332 | **0.578** |
| ZH | rivers | 0.266 ± 0.018 | 0.555 ± 0.019 | 0.154 ± 0.027 | 0.034 ± 0.007 | 0.310 | **0.451** ± 0.021 |
| | cars | 0.344 ± 0.021 | 0.640 ± 0.022 | 0.216 ± 0.040 | 0.019 ± 0.006 | 0.417 | **0.632** ± 0.029 |
| | disasters | 0.355 ± 0.018 | 0.707 ± 0.014 | 0.234 ± 0.029 | 0.031 ± 0.007 | 0.455 | **0.631** ± 0.020 |
| **ZH** | macro avg. | 0.322 | 0.634 | 0.201 | 0.028 | 0.394 | **0.571** |

Table 2: Performance comparison across languages and domains. Cells show bootstrap mean ± half-width for methods with topic-level traces; **SP** is summary-only and has no bootstrap trace. Macro rows average the displayed domain means and therefore omit intervals. Bold marks the strongest deployable condition; grayed-out values are oracle diagnostics excluded from the comparison. **Single**: one generation, no aggregation. **Best samp.**: factuality-best individual candidate, a non-deployable diagnostic. **MC**: response-level majority co-occurrence voting. **MF**: majority fact-level voting. **SP**: SeqProb selection. **LLMAgg**: LLM-based aggregation ($K = 99$).

LLMAgg (0.451) trails the best sampled candidate (0.555) by about 0.104. This is a limitation of the aggregation procedure: the candidate pool can contain a stronger individual response that is not preserved by synthesis.

Bootstrap intervals quantify uncertainty in these descriptive comparisons. They support the reported pattern but are not a formal pairwise hypothesis test; with that caveat, content-level consolidation is more effective here than the evaluated selection and voting alternatives.

**Budget Scaling and the Saturation Point** Figure 2 shows a consistent early-saturation pattern. Factuality improves rapidly as the sample budget increases from very small values, but the marginal benefit declines sharply once the budget reaches a moderate range.

Self-consistency is useful for long-form factual generation: moving from one or two samples to a small multi-sample pool yields large gains in every setting we evaluate. The saturation point, however, is *different across domains and languages* and is not known a priori. EN cars effectively saturates by $K = 8$, EN disasters continues improving through $K = 20$, and ZH rivers shows non-monotonic behavior in this range. This is why we evaluate an input-conditioned policy against a fixed-budget reference.

**Popularity-Tier Analysis** Table 4 shows that popularity remains a strong source of difficulty even under improved inference. Across languages and domains, tail entities are consistently the weakest regime, head entities are generally the strongest, and torso examples lie in between. Self-consistency therefore does not eliminate the long-tail problem.

| Lang | Domain | Retention 0.95 | | | Retention 0.99 | | |
|---|---|---|---|---|---|---|---|
| | | Savings | Retention | Coverage | Savings | Retention | Coverage |
| EN | rivers | +0.061 | +0.997 | +0.964 | +0.046 | +0.997 | +0.971 |
| | cars | +0.287 | +0.994 | +0.850 | +0.135 | +0.994 | +0.914 |
| | disasters | +0.179 | +1.004 | +0.903 | +0.088 | +1.012 | +0.938 |
| ZH | rivers | +0.021 | +1.000 | +0.986 | +0.029 | +0.997 | +0.976 |
| | cars | +0.156 | +1.022 | +0.912 | +0.055 | +1.016 | +0.965 |
| | disasters | +0.115 | +1.001 | +0.936 | +0.062 | +1.000 | +0.954 |

Table 3: Adaptive retention–sampling tradeoff evaluated against the full-budget LLMAgg reference at $K_{full} = 99$. **Savings**: relative reduction in sampled generations. **Retention**: ratio of adaptive to full-budget score. **Coverage**: fraction of examples meeting the retention target. Point estimates are shown.

At the same time, LLMAgg improves every popularity tier, so it is not merely exploiting easy high-popularity cases: the gains extend to the hardest long-tail subset. The size of the gain varies across domains and tiers, but the qualitative pattern is stable. Aggregation therefore helps across the board without closing the head–tail gap: it delivers broad factuality improvements while preserving the benchmark's difficulty structure. Adaptive self-consistency does not remove long-tail difficulty, but it raises factual precision even when knowledge is sparse and entity popularity is low.

**Adaptive Budgeting** Table 3 reports the adaptive retention–sampling tradeoff against the full-budget LLMAgg reference. The adaptive controller achieves score retention close to the full-budget reference across all settings, while reducing the number of sampled generations by $2-29\%$ depending on domain.

The key pattern is heterogeneity. Cars and disasters admit meaningful compression, whereas rivers are much harder to compress aggressively. This

matches the budget-scaling results: when gains saturate quickly, adaptive routing has more freedom to reduce sampled generations relative to the full budget; when the curve is flatter and improvements accumulate more gradually, the room for savings is smaller.

High score retention does not mean uniform per-instance behavior. Coverage is lower in some settings —especially English cars at the looser target —where a non-trivial minority of examples still fall below the retention threshold. Adaptive routing is effective on average, but its payoff depends strongly on input difficulty.

Score retention slightly above 1.0 in some cells is not a contradiction: because aggregation is not strictly monotonic in the number of candidates, a smaller routed budget can occasionally yield a marginally better output than the full-budget reference. Empirically, however, these deviations are small; the main result is that adaptive routing recovers nearly all of the full-budget gain while avoiding unnecessary sampling on many inputs.

**Sensitivity to the Reference Budget** The reported savings depend on the full-budget reference. The budget-scaling results identify $K = 20$ as an empirically near-saturated operating point; against this tighter, calibrated reference, the present pre-generation controller does not produce net savings (see Appendix C for details). The savings in Table 3 should therefore be read as a reduction from the conservative $K = 99$ reference, not as a claim that the controller dominates a hand-tuned saturated baseline. Identifying a saturated baseline requires a budget sweep and may need to be repeated when the model, evaluator, or domain mix changes. Figure 2 reports selected mostly powers-of-two budgets to make the early-saturation pattern readable; the plotted points are exact evaluated prefixes, not interpolated values.

**Qualitative Analysis** The examples in Table 6 clarify the strengths and the risks of aggregation. Successful cases typically arise when correct information is fragmented across several candidates: no individual completion is fully reliable, but the pooled sample set contains enough complementary evidence for LLMAgg to construct a stronger final answer. In this regime, aggregation acts as a synthesis mechanism rather than a selector.

Two error modes recur: repeated hallucination, where the same unsupported content appears across multiple samples and is reinforced rather than removed during aggregation; and entity ambiguity, where multiple plausible referents or unstable interpretations make it hard to identify a consistent factual core. In both cases, adding more samples can actually strengthen the wrong consensus.

These examples explain the empirical results: aggregation is powerful when the candidate pool contains diverse but recoverable factual signal, and much less reliable when the pool is dominated by correlated errors. The same point underlies adaptive routing —additional samples help only when the diversity they introduce contains usable factual information.

| Lang | Domain | Tier | Single | LLMAgg@20 | Δ |
|---|---|---|---|---|---|
| EN | rivers | head | +0.429 | +0.837 | +0.408 |
| | | torso | +0.322 | +0.676 | +0.354 |
| | | tail | +0.160 | +0.363 | +0.204 |
| | cars | head | +0.458 | +0.863 | +0.406 |
| | | torso | +0.450 | +0.815 | +0.365 |
| | | tail | +0.299 | +0.580 | +0.281 |
| | disasters | head | +0.448 | +0.783 | +0.336 |
| | | torso | +0.379 | +0.796 | +0.417 |
| | | tail | +0.270 | +0.596 | +0.326 |
| ZH | rivers | head | +0.510 | +0.749 | +0.239 |
| | | torso | +0.344 | +0.550 | +0.206 |
| | | tail | +0.202 | +0.355 | +0.152 |
| | cars | head | +0.435 | +0.771 | +0.336 |
| | | torso | +0.389 | +0.689 | +0.300 |
| | | tail | +0.295 | +0.559 | +0.264 |
| | disasters | head | +0.495 | +0.750 | +0.256 |
| | | torso | +0.456 | +0.785 | +0.329 |
| | | tail | +0.329 | +0.590 | +0.261 |

Table 4: Performance by popularity tier (head, torso, tail). Δ denotes the gain from **Single** to **LLMAgg@20**. Point estimates are shown; bootstrap intervals are summarized in the Robustness Analysis paragraph.

**Robustness Analysis** The main empirical trends remain stable under bootstrap resampling. Across domains and languages, LLMAgg consistently remains above Single and the weaker baselines, and the budget-scaling curves preserve the same rapid-gain–then-saturation shape. The results therefore look like a stable property of the evaluation setting rather than noise in a few benchmark slices.

The adaptive results are similarly stable. Score retention remains close to the full-budget reference across resamples, while sampled-generation savings retain the same domain-dependent structure discussed above. In particular, the contrast between more compressible domains such as cars and less compressible ones such as rivers persists under resampling, suggesting that this heterogeneity is genuine rather than an artifact of a single split.

For additional context, Appendix B reports corrected ReASC (Kim et al., 2026) results. ReASC is slightly above Single in four of six language–domain pairs, but remains substantially below LLMAgg in all settings. The evaluated selection signals carry some information, but the dominant gains come from content-level synthesis.

**Discussion** Multi-sample inference helps long-form factuality only to the extent that the candidates are combined well. Selection-based methods (SeqProb, the compute-matched best-of-K logprob selector, and the corrected post-hoc ReASC) all improve over single generation, but the gap to LLMAgg is large and stable across languages and domains. Most of this benefit appears early. Budget-scaling curves rise sharply through $K = 8$ and are nearly flat by $K = 16$; across the reported cells, the difference between $K = 20$ and $K = 99$ ranges from $-0.002$ to $+0.015$ (see Appendix B for details). The exact saturation point depends on the domain and language. Within the conservative $K = 99$ reference regime, the pre-generation controller retains nearly all full-budget factuality while reducing sampled generations by 2–29%; against a hand-calibrated $K = 20$ baseline it does not produce net savings (see Appendix C for details).

Long-tail factuality remains the hardest part of the problem. LLMAgg improves all popularity tiers, but does not close the head–tail gap and is not uniformly safe: when the candidate pool is dominated by repeated hallucinations or ambiguous entity references, aggregation can lock in the wrong consensus.

## 6 Conclusion

We presented a study of adaptive self-consistency for long-form factuality across various entity popularity tiers. Across domains and languages, LLM-based aggregation substantially improves over the evaluated selection and voting baselines, including a compute-matched best-of-K selection. Budget-scaling experiments show that most gains appear at moderate budgets, but the saturation point is heterogeneous across domains. Against a conservative full-budget reference corresponding to 99 generations, a pre-generation adaptive policy retains nearly all full-budget factuality while reducing sampled generations by $2-29\%$. We characterize the sensitivity of these savings to the reference budget explicitly. Within the evaluated setting, these results support budget-aware deployment of self-consistency.

## Limitations

Our results are based on a single generator family, so we cannot say from this work alone whether the early-saturation pattern and the rank ordering of aggregation methods carry over to other models. The adaptive policy is intentionally simple and uses pre-generation features. Its savings are measured against a conservative $K_{\text{full}} = 99$ reference; against a tighter, hand-calibrated $K = 20$ reference it does not produce net savings (Appendix C). Closing this gap may require features unavailable before generation, such as partial-output uncertainty or candidate disagreement.

The same appendix discusses two further limitations of the controller: it does not transfer well to held-out domains under leave-one-domain-out training, and it does not allocate noticeably larger budgets to inputs on which LLMAgg degrades performance. The first makes the controller a same-distribution routing policy, not a domain-agnostic recipe. The second leaves catastrophic-failure detection unsolved in this setting; our method does not address it.

Next, multilingual factuality evaluation contains some evaluator noise, which may affect absolute scores on the harder cells. Finally, we did not explore alternative aggregation architectures beyond LLM prompting.

## Use of AI Tools

During the preparation of this work, the authors used LLM assistants to improve the language quality of the manuscript, including grammar and style checking and rewording for clarity. The authors reviewed and edited all content as needed and take full responsibility for the content of the publication.

## References

Pavel Braslavski, Dmitrii Iarosh, Nikita Sushko, Andrey Sakhovskiy, Vasily Konovalov, Elena Tutubalina, and Alexander Panchenko. 2026. The Chronicles of RiDiC: Generating Datasets with Controlled Popularity Distribution for Long-form Factuality Evaluation. *CoRR*, abs/2604.00019.

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal Self-Consistency for Large Language Model Generation. *CoRR*, abs/2311.17311.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *ACM Comput. Surv.*, 55(12):248:1–248:38.

Junseok Kim, Nakyeong Yang, Kyungmin Min, and Kyomin Jung. 2026. Reliability-Aware Adaptive Self-Consistency for Efficient Sampling in LLM Reasoning. In *Findings of the Association for Computational Linguistics, ACL 2026, San Diego, California, United States, July 2-7, 2026*, pages 21575–21590. Association for Computational Linguistics.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When Not to Trust Language Models: Investigating Effectiveness of Parametric and Non-Parametric Memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9802–9822. Association for Computational Linguistics.

Maria Marina, Daniil Moskovskiy, Sergey Pletenev, Mikhail Salnikov, Alexander Panchenko, and Viktor Moskvoretskii. 2026. Boosting Self-Consistency with Ranking. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pages 1017–1032, San Diego, California, United States. Association for Computational Linguistics.

Joshua Maynez, Shashi Narayan, Bernd Bohnet, and Ryan T. McDonald. 2020. On Faithfulness and Factuality in Abstractive Summarization. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 1906–1919. Association for Computational Linguistics.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 12076–12100. Association for Computational Linguistics.

Daniil Moskovskiy, Sergey Pletenev, and Alexander Panchenko. 2024. LLMs to Replace Crowdsourcing For Parallel Data Creation? The Case of Text Detoxification. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, volume EMNLP 2024 of *Findings of ACL*, pages 14361–14373. Association for Computational Linguistics.

Daniil Moskovskiy, Nikita Sushko, Sergey Pletenev, Elena Tutubalina, and Alexander Panchenko. 2025. SynthDetoxM: Modern LLMs are Few-Shot Parallel Detoxification Data Annotators. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 5714–5733. Association for Computational Linguistics.

Andrey Sakhovskiy, Nikita Sushko, Maria Marina, Vasily Konovalov, Elena Tutubalina, Alexander Panchenko, and Pavel Braslavski. 2026. FactOWL: A Cost-Efficient Tool for Long-Form Factuality Evaluation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2026, MelbourneVICAustralia, July 20-24, 2026*, pages 5209–5214. ACM.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters. *CoRR*, abs/2408.03314.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. 2025. Confidence Improves Self-Consistency in LLMs. In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pages 20090–20111. Association for Computational Linguistics.

Llama Team. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Raghuveer Thirukovalluru, Yukun Huang, and Bhuwan Dhingra. 2024. Atomic Self-Consistency for Better Long Form Generations. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 12681–12694. Association for Computational Linguistics.

Artem Vazhentsev, Maria Marina, Daniil Moskovskiy, Sergey Pletenev, Mikhail Seleznyov, Mikhail Salnikov, Elena Tutubalina, Vasily Konovalov, Irina Nikishina, Alexander Panchenko, and Viktor Moskvoretskii. 2026. Leveraging LLM Parametric Knowledge for Fact Checking without Retrieval. *CoRR*, abs/2603.05471.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-Consistency Improves Chain of Thought Reasoning in Language Models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jerry Wei, Chengrun Yang, Xinying Song, Yifeng Lu, Nathan Hu, Jie Huang, Dustin Tran, Daiyi Peng, Ruibo Liu, Da Huang, Cosmo Du, and Quoc V. Le. 2024. Long-form factuality in large language models. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2025. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for LLM Problem-Solving. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Wenting Zhao, Tanya Goyal, Yu Ying Chiu, Liwei Jiang, Benjamin Newman, Abhilasha Ravichander, Khyathi Raghavi Chandu, Ronan Le Bras, Claire Cardie, Yuntian Deng, and Yejin Choi. 2024. WildHallucinations: Evaluating Long-form Factuality in LLMs with Real-World Entity Queries. *CoRR*, abs/2407.17468.

## A Supplementary Material

The current implementation uses the model tokenizer chat template, so the exact serialized prompt depends on the chosen base model. The English and Chinese templates differ only in language and marker string (`Final Consolidated Answer:` and `最终整合后的答案：` respectively).

## B Additional Baselines and Audits

We report two supplementary baselines alongside the main results. The first is a compute-matched best-of-K logprob selector, which picks the candidate with the highest average token logprob from the same 20-sample pool as the paired LLMAgg@K$=$20 result. It is reported here so that the selector is compared at its matched budget, rather than mixed into the full-budget LLMAgg@K$=$99 comparison in Table 2. The second is a corrected version of ReASC, which we audited after an earlier evaluator surfaced anomalous numerical patterns; the bug was that the post-hoc evaluator joined scores at the topic level rather than recomputing them at the completion level from atomic-fact decisions, and once we fixed it ReASC selects the first candidate in only 13–42% of cases rather than collapsing to it. Both baselines are summarized in Table 5. The selector improves over Single in every cell but trails LLMAgg by a wide margin, and the corrected ReASC is slightly above Single in four of six cells but still far below LLMAgg. Score differences between K $=$ 20 and K $=$ 99 for LLMAgg are small throughout, ranging from $-0.002$ to $+0.015$, which is what we mean when we describe K $=$ 20 as near-saturated.

| Lang | Domain | Single | Method | LLMAgg |
|---|---|---|---|---|
| *Best-of-*K *logprob selector (*K$=$20*)* | | | | |
| EN | cars | $+0.344$ | $+0.366$ | $+0.655$ |
| EN | disasters | $+0.287$ | $+0.324$ | $+0.612$ |
| EN | rivers | $+0.211$ | $+0.221$ | $+0.463$ |
| ZH | cars | $+0.349$ | $+0.407$ | $+0.633$ |
| ZH | disasters | $+0.354$ | $+0.417$ | $+0.616$ |
| ZH | rivers | $+0.269$ | $+0.302$ | $+0.442$ |
| *ReASC (corrected post-hoc)* | | | | |
| EN | cars | $+0.344$ | $+0.354$ | $+0.655$ |
| EN | disasters | $+0.287$ | $+0.305$ | $+0.612$ |
| EN | rivers | $+0.211$ | $+0.210$ | $+0.463$ |
| ZH | cars | $+0.349$ | $+0.349$ | $+0.632$ |
| ZH | disasters | $+0.354$ | $+0.368$ | $+0.616$ |
| ZH | rivers | $+0.269$ | $+0.277$ | $+0.442$ |

Table 5: Supplementary baselines. The **Method** column reports either the best-of-K logprob selector (top block) or the corrected ReASC (bottom block), shown alongside Single and LLMAgg (K$=$20).

## C Adaptive Controller: Sensitivity and Failure Modes

The savings reported in Table 3 are computed against the conservative $K_{\text{full}} = 99$ reference. Against the tighter K $=$ 20 reference identified by the budget sweep, the present pre-generation controller does not produce net sampled-generation savings: its average predicted budget across cells ranges from 70.6 to 96.9, well above 20. This indicates that the present feature set does not replace a hand-calibrated saturated baseline. We report savings in sampled generations. The LLMAgg stage remains part of both configurations, but its aggregation prompt grows with K; it is therefore not included in this relative sampled-generation measure.

The controller also generalizes weakly across domains. Under leave-one-domain-out training, coverage drops from in-domain values of roughly $0.85 - -0.99$ to held-out values of $0.24 - -0.65$, and score retention becomes unstable on the held-out slice. In short, pre-generation features capture same-distribution variation but do not transfer reliably across domains, so the controller should be used as a routing policy within its training distribution, not as a domain-agnostic recipe. A related observation is that the controller does not allocate noticeably larger budgets to inputs on which LLMAgg degrades performance: average predicted budgets for loss and win-or-tie groups differ by less than two samples in every EN and ZH cell at both retention targets, and the two ZH catastrophic-loss

cases at the 0.95 target actually receive a lower average budget (57.5) than ordinary win-or-tie cases (90.5). We therefore do not claim to detect catastrophic failures.

Feature importance is consistent with this picture: input length and average word length together account for over 60% of importance, while language and domain indicators contribute less than 3% each (see Figure 3). The controller is reading input complexity, not benchmark structure.

## D RiDiC and FactOWL Details

**RiDiC Benchmark** RiDiC is a benchmark for long-form factuality evaluation designed to assess model performance on entity-centric generation tasks under controlled popularity conditions. The benchmark covers three domains: rivers, cars, and natural disasters, and includes data in two languages: English and Chinese.

RiDiC's defining feature is its explicit stratification of entities into popularity tiers: head, torso, and tail. The tiers are derived from Wikipedia page views and operationalize public entity popularity; they do not directly measure a generator's training-data frequency.

Each instance in RiDiC consists of an entity prompt $x$ and a generated response $y$. The response is a long-form text containing multiple factual statements. This distinguishes RiDiC from short-answer benchmarks, as errors may occur at the level of individual facts within an otherwise coherent response.

Table 1 presents an example from the RiDiC benchmark, illustrating the variability and factual inconsistencies in model outputs.

**FactOWL Evaluation Framework** We evaluate factuality using FactOWL (Sakhovskiy et al., 2026), a framework for fine-grained fact-level verification of generated text. Given a model output $y$, FactOWL decomposes it into a set of atomic factual statements:

$$F(y) = \{f_1, f_2, \ldots, f_n\}. \tag{1}$$

Each fact $f_i$ receives a support status from the evaluation pipeline, resulting in a binary indicator:

$$\nVdash(f_i) = \begin{cases} 1, & \text{if } f_i \text{ is supported,} \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

The factuality score of a response $y$ is then defined as:

$$\text{Fact}(y) = \frac{1}{|F(y)|} \sum_{f_i \in F(y)} \nVdash(f_i). \tag{3}$$

This formulation captures partial correctness of long-form outputs, allowing both correct and incorrect statements to contribute to the final score. The approach is conceptually similar to FActScore (Min et al., 2023), but adapted to the RiDiC setting.

## E Adaptive Controller Features

The controller uses pre-generation features. Prompt-derived features include character, tokenizer-token, and word counts; average word length; digit and punctuation counts; markers such as wh-words, hyphens, and years; and language and domain indicators. A separate domain-level budget prior encodes the near-saturated K from the budget-scaling curve. It is calibration information, not a prompt-derived signal.

We deliberately exclude features that require a model output, such as partial-output uncertainty or inter-candidate disagreement. We therefore describe the policy as pre-generation rather than input-only.

## F LLMAgg Prompt Templates

The prompt bodies below are the exact LLMAgg templates used in the current RiDiC pipeline. The final prompt is assembled with the model chat template, so the effective input includes the system prompt, instruction block, user question, and the candidate answers formatted as `### Answer N`.

We tested minor variations of the aggregation prompt and did not observe material changes in LLMAgg performance, suggesting that the results are not highly sensitive to prompt wording.

## English LLMAgg Prompt Template

**System prompt**
You are an Answer Aggregator. Your task is to read several candidate answers to the same question and produce ONE coherent, high-quality answer.

**Instructions**

- Follow these rules exactly:
- Identify the main points made in each candidate answer.
- Keep points that appear in at least half of the answers.
- Add complementary details that are already present in the existing answers and do not contradict the majority view.
- When answers disagree, choose the version supported by the majority. If tied, pick the clearer or more detailed wording.
- Rewrite everything into a single, well-structured response.
- Do not mention the aggregation process, voting, or the existence of other answers.

**Output format (nothing else):**
Final Consolidated Answer: [your aggregated answer]

**User prompt template**
Question: `{question}`
Candidate Answers (delimited by ###): `### Answer 1 {answer_1} ### Answer 2 {answer_2} ... ### Answer k {answer_k}`
End of input.

## Chinese LLMAgg Prompt Template

**System prompt**
你是一名答案聚合器。你的任务是阅读同一问题的多个候选答案，并输出一个连贯且高质量的最终答案。

**Instructions**

- 严格遵循以下规则：
- 找出每个候选答案中的关键信息点。
- 保留至少一半答案中出现的要点。
- 可以加入候选答案中已有且不与多数观点冲突的补充细节。
- 当答案不一致时，选择由多数支持的版本；若持平，选择更清晰或更详细的表述。
- 将内容重写为一个结构良好的单一回答。
- 不要提及聚合过程、投票或其他答案的存在。

**Output format (且仅输出该格式):**
最终整合后的答案：[你的整合答案]

**User prompt template**
问题: `{question}`
候选答案（以 ### 分隔）：`### Answer 1 {answer_1} ### Answer 2 {answer_2} ... ### Answer k {answer_k}`
输入结束。

| Lang | Domain | Tier | Topic | Single | LLMAgg@20 | Δ |
|---|---|---|---|---|---|---|
| | | | *Wins (LLMAgg improves over Single)* | | | |
| EN | cars | tail | Jiotto Caspita | +0.335 | +0.750 | +0.415 |
| | | head | GAZelle | +0.546 | +0.929 | +0.382 |
| | disasters | torso | Super Outbreak | +0.455 | +0.900 | +0.445 |
| | | torso | Zanclean flood | +0.509 | +0.917 | +0.408 |
| | rivers | head | Blue Nile | +0.513 | +0.938 | +0.424 |
| | | torso | St. Marys River | +0.424 | +0.813 | +0.389 |
| ZH | cars | tail | 劳斯莱斯 Silver Wraith | +0.551 | +1.000 | +0.449 |
| | | tail | 速霸陆 Alcyone | +0.570 | +0.909 | +0.339 |
| | disasters | tail | 2016 緬甸地震 | +0.523 | +0.933 | +0.411 |
| | | tail | 飓风菲乔 | +0.395 | +0.779 | +0.384 |
| | rivers | tail | 馬伊默卡河 | +0.278 | +0.800 | +0.522 |
| | | tail | 巴科伊河 | +0.340 | +0.818 | +0.478 |
| | | | *Failures (LLMAgg degrades performance)* | | | |
| EN | cars | torso | Aston Martin Lagonda | +0.583 | +0.267 | −0.316 |
| | | tail | Audi S1 | +0.819 | +0.450 | −0.369 |
| | disasters | tail | Hurricane Debby | +0.548 | +0.000 | −0.548 |
| | | tail | 2014 Mount Everest avalanche | +0.828 | +0.279 | −0.549 |
| | rivers | tail | Miass | +0.379 | +0.000 | −0.379 |
| | | tail | Tsitsikamma River | +0.674 | +0.235 | −0.438 |
| ZH | cars | tail | 三菱戈蓝拉姆达 / 普利茅斯札幌 | +0.647 | +0.147 | −0.500 |
| | | head | 三菱日蚀 | +0.704 | +0.065 | −0.639 |
| | disasters | torso | 2024 年花蓮地震 | +0.837 | +0.097 | −0.740 |
| | | head | 飓风海伦妮 | +0.788 | +0.000 | −0.788 |
| | rivers | head | 伏尔塔瓦河 | +0.850 | +0.294 | −0.556 |
| | | head | 卢比孔河 | +0.990 | +0.368 | −0.622 |

Table 6: Representative win and failure cases. Δ denotes the change from **Single** to **LLMAgg@20**. Gains typically appear from aggregating complementary evidence across generations, while failures are driven by consistent hallucinations, especially for ambiguous long-tail entities.