When Life Gives You Samples: The Benefits of Scaling up Inference Compute for Multilingual LLMs

Ammar Khairi[1], **Daniel D'souza**[1] , **Ye Shen**[2], **Julia Kreutzer**[1], **Sara Hooker**[1]
[1]Cohere Labs, [2]Cohere
Correspondence: ammar@cohere.com, juliakreutzer@cohere.com

Abstract

Recent advancements in large language models (LLMs) have shifted focus toward scaling inference-time compute—improving performance without retraining the model. A common approach is to sample multiple outputs in parallel, and select one of these as the final output. While existing work has focused on English and specific domains, we study how to robustly scale inference-time compute in a multilingual, multi-task setting: spanning open-ended generations, math and translation tasks, for open models at 8B and 111B scale, across seven languages. Our findings highlight the need for tailored sampling and selection strategies. We propose novel solutions tailored for this multi-faceted inference scenario, demonstrating notable gains across languages and tasks. Our methods achieve an average +6.8 jump in win-rates for 8B models on m-ArenaHard-v2.0 prompts in non-English languages against proprietary models like Gemini. At larger scale, our 111B model shows a +9.0 improvement with just five samples compared to single-sample decoding. These results emphasize the importance of language- and task-aware approaches to democratize inference-time improvements.

1 Introduction

Traditionally, if you wanted higher performance from a machine learning model, you paid for it with more training or data or parameters. A key departure from this is the recent emphasis on scaling up compute at inference time rather than at training time (Wu et al., 2024a; Hooker, 2024; Snell et al., 2025). The combination of growing generative capabilities of large language models (LLMs) paired with better sampling techniques has spurred progress in inference-time compute strategies. These strategies allow for improvements in performance by spending more compute without any alterations to the model itself. However, much remains unknown about how to best search for optimal solutions using inference compute alone, especially for open-ended generative tasks (Zhang et al., 2025b). Even less established is how to tailor inference compute strategies to languages beyond English, which are traditionally under-served by state-of-the-art systems (Üstün et al., 2024; Dang et al., 2024b; Dash et al., 2025), and underrepresented in LLM research.

In our work, our goal is to understand how to *most robustly scale inference compute across languages for generative tasks*. For a given model, how can we best invest a fixed budget of inference-time compute to improve performance across all languages? We are most interested in techniques that generalize across open-ended tasks and formally verifiable tasks, and across languages. Hence, our setting is *extremely multi-task* with many different performance constraints to balance.

We focus on *parallel scaling* (Wang et al., 2023; Welleck et al., 2024; Zhang et al., 2025b) which increases inference-time compute by first generating multiple outputs in parallel and then selecting one of them as final output.[1] We can think of parallel scaling as an endeavor to make the best lemonade from an already grown lemon tree (the trained model): First we carefully harvest lemons (generate samples) and then pick the best of them (selection) for the lemonade. These two stages, and how well they are aligned, determine the power of inference scaling (Stroebl et al., 2024; Brown et al., 2024; Huang et al., 2025). Our work shows that we have to depart from go-to solutions for English to account for language-specific variance. Our primary contributions are the following:

1. **Extensive study of existing methods.** There are many prior approaches to the problem (Ippolito

[1]This idea is known under many other names, e.g. Best-of-N (Huang et al., 2025), "repeated-sampling-then-voting" (Chen et al., 2025), or rejection sampling (Touvron et al., 2023)

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 27559–27583
November 4-9, 2025

et al., 2019; Shi et al., 2024; Snell et al., 2025), such as Best-of-N (BoN) selection with a reward model (RM), or Minimum Bayes Risk (MBR) decoding with pairwise LLM judgments, but they have studied subsets of tasks and languages. We empirically investigate sampling and selection inference strategies under our massively multi-task constraints, spanning two multilingual LLMs, seven languages and three generative tasks (open-ended generation, math, machine translation).

2. **Novel risk-reducing sampling.** We introduce a novel **hedged sampling** method to better exploit the diversity obtained through sampling under increased softmax temperature. It specifically benefits languages that come with a higher risk of sample quality deterioration at high temperatures. Opposed to prior works that recommend either stochastic or deterministic sampling depending on the task (Song et al., 2025), we show that *including both in the sample set is key to multilingual generalization*.

3. **Improved selection strategies.** We propose two new selection strategies which capitalize on *long-context modeling and the versatility of cross-lingual generation abilities* of recent generalist LLMs. We call these **Checklisted One-Pass Selection (CHOPS)** and **Cross-lingual MBR (X-MBR)**. Using only five samples, we show that they bring +17.3% (AYA EXPANSE 8B), +9.4% (QWEN3 8B), +9.0% (COMMAND A) increase in win rates on multilingual ArenaHard v2.0 compared to single samples, often outperforming BoN with specialized RMs.

We distill our findings into a *recipe for squeezing the most out of multiple samples in a multilingual and multifaceted generation paradigm*, which we coin the "Multilingual LLMonade Recipe". Our findings have implications for the broader test-time scaling landscape, as they demonstrate that careful design of sampling and selection techniques can bring important gains even at the low-end scale of inference-time scaling for high-end multilingual LLMs. Contrary to the trend of exploiting specialized RMs for single-task inference-time scaling, generalist LLM judges bring robust improvements even in challenging and diverse multi-task setups, thanks to their versatility and adaptability.

Multilingual LLMonade Recipe

Step 1: Use **hedged sampling** to generate N samples.
Optional: Localize a reasonably high temperature (start τ at 0.7–0.9) for different contexts.

Step 2: Use a **multilingual LLM to select** the best sample, using CHOPS or X-MBR with auxiliary samples from a dominant language.
Optional: A small exploration of multilingual LLM judges to find the best suited one.

In the following, we take apart the question on how to optimize multi-sample inference, by first investigating the *sampling strategy* (section 2), then comparing multiple *selection strategies* (section 3). We mark our newly introduced methods and their empirical effects with 🍋, and contrast English results (●) with non-English (◎) results.

2 How to Sample?

The first ingredient for successful test-time scaling is the creation of a sample pool of sufficient quality: At least one sample in the pool needs to be of higher quality than what can be expected from a single sample. Our research question is here: *How to create a sample pool with a strategy that is robust across languages and tasks?*

2.1 Methodology

Temperature Sampling Our first task is to create a valuable pool of generations via stochastic sampling. We explore different variants of temperature sampling (Ackley et al., 1985) as it offers an intuitive way of steering the diversity and quality of the generation pool. Temperature sampling divides the logits for each token prediction by a fixed constant $\tau > 0$: $\texttt{softmax}(l_\theta/\tau)$. Under high temperatures ($\tau > 1$) the resulting probability distribution becomes more uniform, while at its extreme (close to 0), it becomes more unimodal. As a consequence, higher temperatures generate a more diverse pool of samples. The quality, however, varies depending on the task, language, and model. At 0, sampling becomes *greedy* decoding, picking the token with the maximum likelihood at each decoding step, which we refer to as $\tau = 0$ for simplicity.

🍋 **Multi-Temperature Sampling** Prior works investigated sampling the entire pool from the same temperature (Du et al., 2025; Renze, 2024; Song et al., 2025). When we have a large variety of

tasks and languages, representing very different subspaces of the data distribution that the model is trained on, it might be impossible to set the temperature optimally for all. For inputs that the model is well trained on, higher temperatures can be tolerated without incurring loss of quality, while for lesser trained inputs, lower temperatures are required to maintain quality. Alignment also plays a role in that it stabilizes robustness under higher temperatures for the inputs that the model sees during alignment (Shi et al., 2024). To increase robustness, we thus investigate composing the sample set from outputs generated under mixed temperatures $\tau \in [0, 1]$. These temperatures can either be chosen randomly, or according to insights from a development set. As we will show below, temperature sensitivity varies across languages and tasks, so random sampling from a reasonable range is better than choosing just one potentially suboptimal temperature for all settings. Tuning temperatures individually per task and language is unrealistic due to combinatorial explosion, especially with massively multilingual and multitask goals.

🍋 **Hedged Temperature Sampling** As a variation of multi-temperature sampling, we propose a *hedged sampling* strategy which additionally includes deterministic outputs from greedy search ($\tau = 0$) in this mix. This helps hedge risk because even if there are lower-quality samples due to variance caused by higher temperatures, we can default back to strong deterministic sample candidates for aligned models (Song et al., 2025). Hedged sampling is complementary to token-level techniques that truncate the output space of individual stochastic samples, e.g. by pruning low-probability tokens with a fixed threshold (ϵ-sampling) (Hewitt et al., 2022; Freitag et al., 2023), or a dynamic threshold based on the model's confidence (min-p sampling) (Minh et al., 2025). Hedging risks is particularly important for multilingual applications, where the risk tends to be higher in less dominant languages, as we will show in the following experiments.

2.2 Experimental Setup

Multilingual Multi-tasking Open-ended generation tasks have received less attention in test-time scaling works. It is harder to fit a single method or reward model to the diverse challenges that open-ended generations pose. Our goal here is to take a wider view, which means considering both open-ended tasks and tasks with underlying correctness.

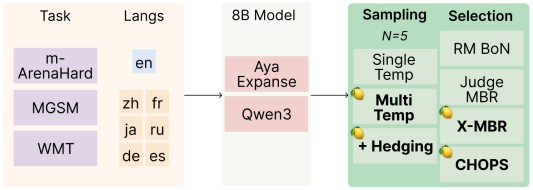


Figure 1: Overview of the **multilingual multi-task** experimental scope. New methods are marked with 🍋.

The experimental setup is summarized in fig. 1. We test AYA EXPANSE 8B (Dang et al., 2024b) and QWEN3 8B (Yang et al., 2025a) models on 7 languages, selected for their inclusion in our target benchmarks. We evaluate on three task types:

1. **Open-ended generation (Arena)**: We measure win rates on m-ArenaHard v2.0[2] using GPT-4o (`gpt-4o-2024-05-13`) as judge, comparing against greedy decoding for intrinsic, and Gemini (`gemini-2.0-flash`) for extrinsic comparison.

2. **Mathematical reasoning (MGSM)**: We measure accuracy[3] on MGSM (Shi et al., 2022).

3. **Machine translation (WMT)**: We measure translation quality from English with XComet-XL (Colombo et al., 2023) on collections of the yearly WMT task (Federmann et al., 2022; Deutsch et al., 2025).

To properly measure generalization, we use separate data splits for development and testing, described in table 6, together with full experimental details in appendix A.

Budget Size for Parallel Scaling Compared to prior works that investigate sample sizes in the hundreds to thousands (Freitag et al., 2023; Song et al., 2025; Huang et al., 2025), we focus on the lower end of inference-compute scale. We set $N = 5$, given that it is a more realistic workload for large scale production systems (i.e. many inputs applied with 5× the amount of the normal compute). Complementing this view, we also observe that scaling curves tend to have their steepest incline in the first steps, i.e. the highest return for additional invested compute, especially for imperfect selection methods (Brown et al., 2024; Chen et al., 2025). In

[2]Released at https://huggingface.co/datasets/CohereLabs/m-ArenaHard-v2.0

[3]We use `simple-evals`'s exact match metric https://github.com/openai/simple-evals/tree/main

Appendix E.4, we report similar findings on our generative tasks when increasing N up to 40: An even larger inference budget does not necessarily translate to much higher performance when looking at metrics like win rates across languages.

Measuring Sample Pool Quality In order to compare the effects of parallel scaling across tasks, we introduce HOPE and RISK metrics, which intuitively represent the hope and risk of scaling up compared to the quality of a single sample. HOPE is defined as the relative change between the score of the best sample in the set $y^+ = \arg\max_{y \in Y} r(y)$ and the evaluation score of the greedy output $\hat{y}$: $\frac{r(y^+) - r(\hat{y})}{r(\hat{y})}$. For RISK, we analogously compare the relative change in evaluation score between the worst sample y^- and the greedy output. For open-ended generations, we query an in-house multilingual reward model that scores an average RewardBench score of 76.1 on RewardBench2 (Malik et al., 2025) to estimate the quality of individual samples, while for the other tasks we evaluate the samples against the reference with the respective task metrics. We report HOPE and RISK averaged across instances from each benchmark.

2.3 Results

Higher temperature sensitivity for non-English. In classic single temperature sampling, we dedicate our entire inference budget to sampling at one fixed temperature. Figure 2 compares how best, worst and average performance differ across all three tasks if we spend this budget at different temperatures. We observe consistent trends: As temperature increases, the gap between best-case and worst-case outcomes widens. While higher temperatures lead to improved best-case scenarios, they also increase the chances of generating lower-quality examples. Notably, the rate at which variance increases is influenced by both the language and the nature of the task, likely influenced by their presence in training and alignment of the underlying model. Comparing English against the other languages, its average sample quality is more stable even at higher temperatures (more *eurythermal*), while it drops earlier for other languages. Likewise, best-sample quality continues to grow till close to $\tau = 1$ for English, while it decays earlier and steeper for other languages (here Japanese, more languages in appendix C).

Higher risk and hope for non-English. Table 1 quantifies the HOPE of quality increases and RISK of quality drops of repeated sampling at $\tau = 0.7$. First, the HOPE results highlight the significant headroom achievable with just five samples if selected optimally. Second, we can see that non-English (◎) sampling comes with higher HOPE (37.2%) but at greater RISK (–44.5%) than English (●). This illustrates the importance of balancing potential gains against possible losses in parallel sampling, particularly in multilingual settings where it is more risky to sample at higher temperatures.

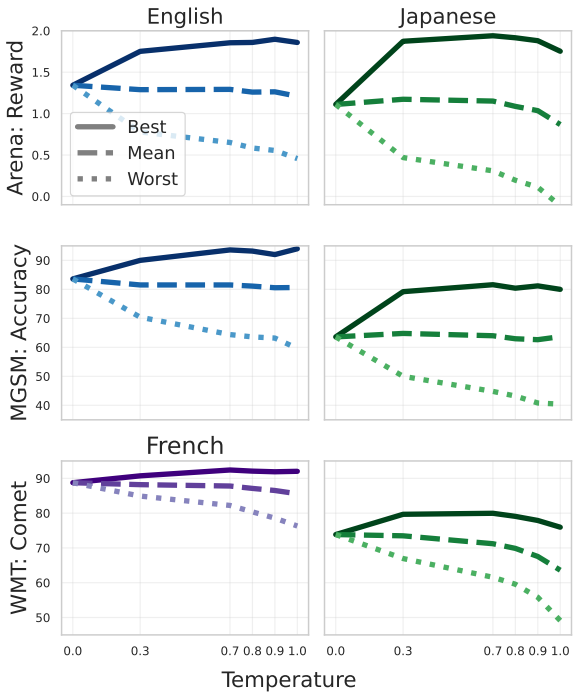


Figure 2: **Quality under single temperature sampling** We evaluate best, worst and mean quality for $N = 5$ samples from AYA EXPANSE 8B for English (left, replaced by French for WMT) and Japanese (right) on each of the dev sets of the tasks (rows: Arena, MGSM, WMT).

Setting	Task	HOPE	RISK
● English	Arena	38.81	-51.49
	MGSM	11.90	-23.81
	Average	**25.36**	**-37.65**
◎ Non-English	Arena	55.51	-65.31
	MGSM	18.96	-23.70
	Average	**37.23**	**-44.50**
◎ Non-English	WMT	6.47	-11.64

Table 1: **HOPE and RISK** of sampling at $\tau = 0.7$, relative to evaluation scores of greedy decoding for AYA EXPANSE 8B. Values are reported as percentages.

Greedy outputs are the best single-sample bet. When inspecting the mean scores of samples from different temperatures in Figure 2, it becomes clear that the quality of greedy outputs ($\tau = 0$) is always greater or equal than the *expected quality* of outputs sampled at higher temperatures. This is consistent across setups, corroborating the recommendation for greedy decoding by (Song et al., 2025). Therefore, greedy decoding is our single-sample baseline for measuring the benefits of scaling up.

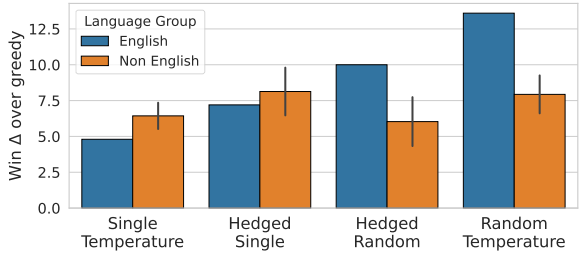


Figure 3: **Single vs. multiple temperature sampling, with and without hedging**: Win rates over greedy outputs on mArenaHard with Judge MBR and $N = 5$ samples, averaged across models. Error bars show variance across non-English languages. For single temperature, we use $\tau = 0.7$; for multiple temperatures, we sample uniformly from $\tau \in \{0.0, 0.3, 0.7, 0.8, 0.9, 1.0\}$. Hedging replaces one sample with the greedy output.

🍋 **Multi-temperature sampling benefits English** Instead of attempting to tune temperatures for each possible inference scenario, we use our budget to sample across multiple temperatures, ($\tau \in [0.0, 0.3, 0.7, 0.8, 0.9, 1.0]$) by randomly picking $N = 5$ values from this set uniformly without replacement. We compare this against sampling from a single temperature ($\tau = 0.7$, highest best-case performance on average across setups, see Appendix E.1). To measure downstream effects, we utilize Judge MBR as a selection technique (details will follow in section 3) and measure m-ArenaHard win rates for all sampling techniques against greedy outputs. Figure 3 confirms that there are indeed downstream benefits for random multi-temperature sampling compared to single-temperature sampling. However, these gains are most pronounced for English +8.8% (compared to Non-English with +1.4%), because other languages have a higher RISK is at high temperatures (fig. 2) included in the set.

🍋 **Hedged sampling is best for non-English** We now test if *hedging* with a greedy sample in the pool can help manage the higher RISK across languages at high temperatures. Figure 3 visualizes the effect of hedging for m-ArenaHard win-rates over greedy outputs. When we combine 4 samples at $\tau = 0.7$ with the greedy output for the MBR method to choose from ("Hedged Single"),[4] we find that this safety net effectively increases win-rate deltas by +1.73 in Non-English languages. This constitutes the largest improvement over single temperature sampling in non-English languages. Overall, single temperature sampling with hedging balances English and non-English performance best, so we choose it as a base for our selection experiments.

Combination with probability pruning Hedged sampling can further be combined with techniques that reduce risk at the token level, such as min-p (Minh et al., 2025). In our setup, min-p provides additional gains over hedged sampling alone in most test cases, see appendix E.2. The improvements are most consistent for machine translation, where similar probability pruning techniques have previously been shown essential for MBR (Freitag et al., 2023). We apply min-p in final test set evaluations in the next section and denote this combination with the subscript min-p.

3 How to Select?

Once we have sampled a pool of generations of promising quality, our goal is to correctly identify the best generation in the pool. The research question is: *How to select from a sample pool with a strategy that is robust across languages and tasks?*

3.1 Methodology

We briefly review multiple selection techniques of varying complexity, and propose our own extensions (🍋) that are particularly equipped for multilingual generative tasks.

Maximum Likelihood Given a pool of samples Y, the sample $\hat{y}$ with the highest likelihood under the model distribution $p_\theta(y \mid x)$ should be a good candidate for selection when the model is well calibrated (i.e. likelihood and quality correlate): $\hat{y} = \arg\max_{y \in Y} p_\theta(y \mid x)$. This constitutes an intrinsic metric that relies only on the model.

Best-of-N (BoN) introduces an *extrinsic utility metric* $U(y)$ to score each sample independently. The sample with the highest utility score gets selected: $\hat{y} = \arg\max_{y \in Y} U(y)$. This approach re-

[4]The analysis in appendix C shows that MBR selects greedy for 35.3% of prompts on average.

lies on the utility metric being well aligned with the task evaluation metric and well calibrated, i.e., rating outputs adequately on a common scale even if scored independently. BoN is typically used with specialized reward models (RM BoN) (Zhang et al., 2024a; Ichihara et al., 2025; Pombal et al., 2025; Son et al., 2025) or verifiers in math or code tasks (Snell et al., 2025; Cobbe et al., 2021; Lightman et al., 2024; Zhang et al., 2025a).

Minimum Bayes Risk (MBR) decoding searches for the candidate $\hat{y}$ that *minimizes the expected risk* over the distribution of samples (Kumar and Byrne, 2002, 2004; Eikema and Aziz, 2020, 2022). The risk $R(y')$ of a candidate y' is approximated with pairwise comparisons from the sample pool: $R(y') \approx \frac{1}{|Y|}\sum_{y \in Y} L(y, y')$ where $L(y, y')$ is a pairwise loss function measuring the discrepancy between a candidate y and and a pseudo-reference y'. MBR thus selects

$$\hat{y} = \arg\min_{y' \in Y_h} \sum_{y \in Y_e} L(y, y'), \tag{1}$$

where $Y_h \subseteq Y$ is the hypothesis set, and $Y_e \subseteq Y$ is the evidence set used to estimate the risk. As Bertsch et al. (2023) highlighted, the evidence set Y_e aims to cover a representative portion of the space for accurate risk estimation, while the hypothesis set Y_h focuses on the narrower, high-quality region to avoid considering low-quality candidates, but they do not need to be identical.

For loss functions, there are many possible implementations: When aligned well with the task evaluation metric, it *reduces the optimization-evaluation gap* and brings larger empirical gains (Kovacs et al., 2024), which has made it a popular method in machine translation and open-ended generation (Fernandes et al., 2022; Freitag et al., 2023; Wu et al., 2025). In this way, we can optimize for pairwise comparisons under an LLM judge at test time, such as win-rate evaluations. When loss functions focus on similarity (e.g. token-based similarity), MBR becomes the equivalent to majority voting in classification tasks (Bertsch et al., 2023). It selects the sample that is most *consistent* with the evidence set, which relates it to the notion of self-consistency (Wang et al., 2023; Shi et al., 2024; Chen et al., 2025; Wang et al., 2025).

Checklisted One-Pass Selection (CHOPS) Most of the prior selection methods present considerable computational costs: BoN requires N model calls, and MBR even N^2 due to pairwise comparisons. This may be a reasonable approach for some latency-insensitive tasks, but we pursue an alternate approach that reduces this efficiency penalty. Capitalizing on the development of longer context windows for LLMs, we prompt the model to generate a checklist to help it then *choose one of the presented samples* (see appendix D). This is inspired by the success of rubrics to facilitate LLM judge decisions (Kim et al., 2024) and the ability of LLMs to generate prompt-specific checklists (Cook et al., 2024), which help to adapt the judge on-the-fly to diverse selection scenarios across languages and tasks. CHOPS requires only one model forward pass, fitting all samples into the input context at once. An ablation in appendix E.3 confirms that self-generated checklists are a essential component, especially for non-verifiable tasks and non-English.

Crosslingual MBR (X-MBR) Orthogonal to the motivation to reduce LLM judge calls, we propose a second selection method X-MBR, that is motivated by strong crosslingual transfer abilities of LLMs. Building directly on the MBR paradigm, X-MBR uses cross-lingual evidence to more robustly select from target-language candidates. We hypothesize that it will be easier for the judge to select the best target language sample when it is presented with higher-quality samples from other (more dominant) languages. For an input x, X-MBR uses the same *hypothesis set* as standard MBR, i.e. the same five samples in the target language. The novelty lies in the *cross-lingual evidence set* Y_{ex}, that extends original in-language evidence set Y_e from eq. (1) by a smaller set of cross-lingual samples ($M < N$). These samples are generated by instructing the same LLM to respond in different "evidence" languages (e.g. "Answer in English", see prompt in listing 2). Note that this does not require prompt translation, it solely relies on cross-lingual generation. We then pick the candidate from the hypothesis set that accumulates the highest cross-lingual support, with the same MBR selection criterion as for classic MBR (eq. (1)), but including additional cross-lingual comparisons by the LLM judge between the hypothesis set Y_h and the set of new cross-lingual samples Y_{ex}:

$$\hat{y} = \arg\min_{y' \in Y_h} \sum_{y \in (Y_e \cup Y_{ex})} L(y, y'). \tag{2}$$

This exploits both the cross-lingual generation abilities of the model that we sample from, as well as

the cross-lingual comparison abilities of the judge LLM. X-MBR requires generating $N + M$ total samples and performing $N \times (N + M)$ pairwise comparisons, approximating $\mathcal{O}(N^2)$ complexity. It is an interesting direction which explores the return on additional compute investment through cross-lingual validation.

3.2 Experimental Setup

Baseline We compare against **greedy** decoding, which in each generation step selects the token with the highest model probability, resulting in deterministic and predictable output. This gives us a simple yet effective comparison point (Song et al., 2025), as we have empirically confirmed in section 2. In this way we can quantify the benefits from scaling the generation budget from 1 to 5 samples.

Sampling Strategy To ensure consistency in comparisons since our goal is to isolate the best selection techniques, all methods operate on the same pool of $N = 5$ samples generated with hedged sampling at $\tau = 0.7$.

Choice of Utility Metrics Some of the above selection methods can be used with multiple utility metrics or backbone models, such as BoN or MBR. In order to disentangle the effect of the method from the underlying utility metric, we compare multiple instantiations of each. Concretely, we benchmark two versions of MBR, **2-Shingle MBR**—which relies on the simple Jaccard similarity of token-level bigrams (2-Shingles) from pairs of generations[5], and **Judge MBR**—which queries an LLM judge for pairwise comparisons. In preliminary experiments, we also explored using the LLM judge for BoN vs the RM, but the LLM judge did not perform competitively due to missing calibration for generating absolute scores.

Choice of RM and Judge Model Based on prior findings that the precision of the utility score can have major impact on the success of inference-scaling (Huang et al., 2025; Stroebl et al., 2024), we aim to pick the best scoring open judge or RM model for our experiments. For techniques which use an LLM judge, we use `Command A` (Cohere et al., 2025), an 111B model optimized for multilingual performance supporting 23 languages. `Command A` is scoring competitively to `GPT-4o` on mRewardBench (Gureja et al., 2024). For BoN RM, we choose the leader from the RewardBench leaderboard (Lambert et al., 2024) `INF-ORM-Llama3.1-70B` (Minghao Yang, 2024), which is based on the multilingual Llama3.1 base. It is trained on a mix of open-sourced preference pairs (Liu et al., 2024) with difference magnitudes determined by `GPT-4o` (Wang et al.). This RM is a very strong competitor for any LLM judge approach, it is specifically engineered and trained to perform generation scoring aligned with `GPT-4o`. Appendix B details the selection process of RM and LLM judge. Each selection method with LLM judge requires a specific prompt, which we list in appendix D. For consistency, the instruction prompts are always in English.

[5] https://nlp.stanford.edu/IR-book/html/htmledition/near-duplicates-and-shingling-1.html

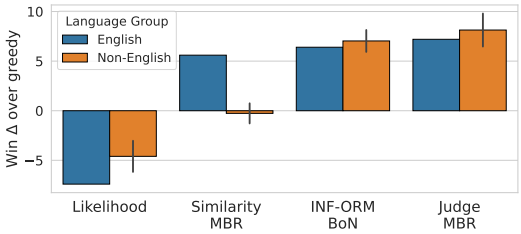


Figure 4: mArenaHard win-rate comparisons to greedy outputs for **naive methods vs. RM- and LLM-based approaches** that select from 5 samples, averaged across models. Error bars: variance across languages.

3.3 Developing the Best Selection Strategy

Naive approaches are not competitive. Figure 4 compares selection methods that do not rely on LLM judges or RMs with those that do. We report averages across models and languages with error bars to show the variance between different languages. A full breakdown for the baseline methods and judge based methods (with significance measures) are included in Table 13 and Table 12, respectively. We observe that Maximum Likelihood selection results in losses across the bench, suggesting that the model's internal probabilities are not calibrated for win rate evaluation. Similarity-based MBR, which selects the sample most similar to others in the pool, only yields improvements in English (●+*5.6%*) but performs comparably to greedy in non-English ◎. We suspect this is due to the higher variance of quality for the samples in the pool for non-English (higher RISK), which does not sustain picking the most consistent sample.

BoN & judge-based selection outperform greedy. RM BoN shows consistent improvements for both groups (●+*6.4%*, ◎+*7.0%*), establishing it as a

	Model	MBR	S-MBR	En-MBR	Zh-MBR	R-MBR
●	AYA	11.20	12.00	–	**15.20**	13.20
	QWEN	3.20	9.20	–	**20.00**	4.40
	Avg	*7.20*	*10.60*	–	***17.60***	*8.80*
◎	AYA	8.13	13.27	**15.93**	12.64	11.53
	QWEN	8.13	6.07	**9.87**	8.40	7.53
	Avg	*8.13*	*9.67*	***12.90***	*10.52*	*9.53*
Overall Avg		*7.67*	*10.13*	*12.90*	*14.06*	*9.17*

Table 2: **Expanding MBR evidence sets:** Comparison of different sources of evidence for 3 additional samples on m-ArenaHard: MBR: No additional samples, S: from the same language, R: random samples across languages, En: from English, Zh: from Chinese.

strong baseline. This corroborates recommendations by Wu et al. (2024b) to use reward models crosslingually for BoN when they have a multilingual LLM backbone. Judge-based MBR achieves the highest deltas (●*+7.2%*, ◎*+8.1%*), showing that a general-purpose multilingual LLM judge can outperform a specialized reward model. The flexibility of using LLMs for pairwise comparisons in the MBR setup aligns well with pairwise win rate evaluations, but it requires N^2 comparisons.

🍋 **Better judgment with crosslingual evidence.** We extend the evidence set of MBR with $M = 3$ more samples, either (1) samples of the same language (S-MBR), (2) a fixed language (English, Chinese; En/Zh-MBR),[6] or (3) randomly chosen languages for all prompts (R-MBR). In table 2, we compare these variants, all using the same candidates for selection and only differ in the composition of the evidence set. S-MBR results in a slight improvement of win rates (from +7.7% to +10.1%). Sampling from random language (R-MBR) yields similar results, indicating that there is no crosslingual benefit. Key to success is sampling evidence from dominant languages: Both En-MBR and Zh-MBR result in significant improvement over both greedy and the S-MBR baseline, particularly in non-English languages (◎). This shows that we can effectively leverage the model's multilingual capabilities to enhance performance across all languages. It underscores the potential of multilingual LLM judges to optimize available test-time compute.

[6]We choose English and Chinese as instances of high-resource languages that we assume are dominant languages for both models.

Task	Model		RM BoN$_{\text{min-}p}$	CHOPS$_{\text{min-}p}$	X*-MBR$_{\text{min-}p}$	*Greedy*
Arena	AYA	●	**19.60**	14.40	16.80	–
		◎	16.27	**17.33**	15.67	–
	QWEN	●	2.00	7.60	**8.80**	–
		◎	5.87	8.27	**9.40**	–
MGSM	AYA	●	7.76	6.96	**7.76**	77.84
		◎	**9.59**	6.19	7.92	69.55
	QWEN	●	**3.04**	1.84	0.64	94.96
		◎	3.65	2.19	**3.85**	84.41
WMT	AYA	◎	**1.04**	0.72	0.20	71.92
	QWEN	◎	**1.43**	1.12	0.93	76.15

Table 3: **Test set results**: Quality gains over greedy decoding by selecting from five samples (🍋 hedged $\tau = 0.7$ and min-p with $p = 0.2$). X*-MBR uses Chinese as evidence languages for English, and English for the rest.

Arena		RM BoN	CHOPS	X*-MBR	*Greedy*
AYA vs Gemini	●	**7.60**	2.80	3.60	20.80
	◎	6.87	5.07	**7.33**	25.80
QWEN vs Gemini	●	1.60	**2.80**	0.40	38.40
	◎	6.00	5.00	**6.33**	42.07

Table 4: **Open ended test results with extrinsic comparison**: Gains in win rates against GEMINI 2.0 FLASH.

3.4 Testing in Multilingual Multitasking

Bringing it all together: Test set performance. Table 3 compares test results for the LLM-judge based aggregation methods based on hedged sampling with $\tau = 0.7$ and min-p with $p = 0.2$ across tasks and models. We find improvements over the greedy baseline (i.e., the best single-sample method) in all tasks and languages with magnitudes of improvement that are substantial; considering that we are working with as few as five samples.

🍋 **BoN vs CHOPS vs X-MBR** For open-ended in Table 3, both CHOPS and X-MBR outperform RM BoN selection in most cases, with e.g., CHOPS getting up to +17.3% (◎) improvements in win rates on Arena compared to the greedy sample. In table 4, we additionally compare them against one sample from the larger and more capable GEMINI 2.0 FLASH model. Even under this adversarial comparison, gains for all three approaches are notable, with X-MBR achieving the highest non-English gains with an average of +6.8%. In close-ended evaluation, we find significant gains across all methods, even with strong greedy performance. For MGSM, X-MBR achieves the highest gains for non-English QWEN, +3.9% accuracy, where larger cross-lingual performance gaps provide more room for improvement. RM BoN performs particularly well on WMT translation tasks with a +1.2 gain in

Model		RM BoN	CHOPS	X*-MBR
COMMAND A	•	1.60	6.00	**7.60**
	◎	7.47	**10.73**	10.40

Table 5: **Self-improvement with parallel scaling:** COMMAND A generates samples and selects from them. Win rates gains over the greedy single sample baseline.

XComet scores. Overall, both our proposed selection methods yield substantial gains, but CHOPS might in practice be more attractive since it requires less compute.

Successful extension to self-improvement. One might think that the benefits of LLM judges for selection vanish when there is less of a capability gap between sampling model and selecting model (so far 8B vs 111B). Therefore, in Table 5, we evaluate a *self-improvement* scenario where we use the 111B COMMAND A to both generate samples and perform selection for CHOPS and X-MBR. Not only do we find consistent gains across languages, but our new selection methods outperform RM BoN, with CHOPS and X-MBR obtaining remarkable deltas in Non-English ◎of +10.7% and +10.4%, respectively, compared to RM BoN's modest +7.5%.

4 Related Work

Stochastic vs Deterministic Inference Early LLM research suggested that diversity through stochastic inference often comes at a cost of quality (Holtzman et al., 2020), benefiting some tasks while hindering others (Holtzman et al., 2020; Peeperkorn et al., 2024; Renze, 2024). Song et al. (2025) found that closed-ended and verifiable tasks favor deterministic decoding, while open-ended generation benefits from stochastic sampling. Du et al. (2025) use entropy measures to optimize the temperature(excluding $\tau = 0$) across math and coding tasks in English. In our work, we focus on exploiting the variance in the smaller sample range ($N = 5$) across *multiple generative tasks*. We also add the dimension of language that has previously been ignored: *Going beyond English* and address variance in higher temperature samples by hedging it with deterministic inference.

Multilingual Test-time Scaling and Alignment While most test-time scaling and alignment research focuses on English, a few recent works have explored multilinguality. Pombal et al. (2025) propose a multilingual judge LLM for BoN, showing improvements in win rates across three languages. Gupta and Srikumar (2025) also confirm the potential benefits of RM BoN for multilingual open-ended tasks across various model and sample sizes. Our study expands on these previous RM BoN explorations with a broader set of tasks and novel methods, specifically crafted for the challenges in multilingual generation both on the sample generation and the selection side. What emerges as a consistent pattern in their and our work is that RMs appear to generalize well across languages for parallel scaling, even if trained only on rewards for English (Wu et al., 2024b). Similarly, Yong et al. (2025) demonstrate cross-lingual scaling benefits in math and STEM reasoning with a LLM with a multilingual backbone tuned for English reasoning. They show benefits of non-target language reasoning/scaling, which is loosely related to the effectiveness of crosslingual evidence for X-MBR that we find in our experiments. With the shared motivation to reduce imbalance across languages, Yang et al. (2025b) and Zhu et al. (2024) use cross-lingual sample generation with a translation pipeline, while Yoon et al. (2024) combine expert models for task and language expertise. Our X-MBR approach achieves significant gains without intermediate translation or experts, leveraging the LLM's cross-lingual generation capabilities.

5 Conclusion

We have conducted extensive experiments on three generative tasks to compile a recipe for multilingual parallel scaling that generalizes across both tasks and models. Based on our insights on the impact of temperature on sample pool quality, we designed a hedged temperature sampling variant, and combine it with selection methods tailored towards multilingual judges. We propose two approaches which improve upon existing methods: Checklisted One-Pass Selection (CHOPS) and Cross-lingual MBR (X-MBR). These techniques show consistent cross-lingual gains in the benefits of test-time scaling. This has not only implications for inference, but also for applications where multilingual inference is an intermediate step in model improvement, e.g. for synthetic data generation (Thakur et al., 2024; Dang et al., 2024a; Odumakinde et al., 2024) or distillation (Zhang et al., 2024b), test-time alignment (Sun et al., 2024; Amini et al., 2025) or model fine-tuning (Touvron et al., 2023; Snell et al., 2025).

Limitations

Reliance on judge alignment All methods that use extrinsic signals (reward models or LLM judges) for selecting from multiple sample are bounded by their alignment with the evaluation metric, as has previously been pointed out in (Stroebl et al., 2024; Huang et al., 2025). Our methods do not directly address this issue. By selecting the latest and most generalist judge models for selection, we hope that the effects of task-specific reward hacking / mismatch are reduced.

Language selection Our selected languages are all high-resource languages and well represented throughout the stages of LLM training. Our study does not cover the test case of generalizing to underrepresented languages that are unsupported by the model or not included in stages beyond base model training. We can expect that both quality of samples and LLM aggregation precision will be significantly lower, so approaches like X-MBR that leverage crosslingual knowledge might be more promising.

Sample scale As explained in section 2, we focus on the low-end of test-time scaling in terms of sample sizes, and prioritize spending compute on potentially expensive selection methods. Scaling up further $N > 5$ might be interesting for further pushing the limits, but we instead focus on making the most of few samples that already give us substantial headroom. Scaling up N further poses different challenges for bridging misalignments between selection methods and e.g. win rates, see appendix E.4, so it might not be the most promising investment of additional compute.

Cost of selection method We found that a larger generative model was needed to improve upon the base model performance (based on preliminary explorations with mPrometheus (Pombal et al., 2025)). In practice, it is more attractive to employ a smaller judge model so that it does not dominate the added inference cost. One solution would be to distill the outputs of the large generative judge into a smaller model.

Acknowledgments

We thank our colleagues at Cohere and Cohere Labs for their help in refining the paper, in particular: Sander Land for the reward model scoring, Thomas Euyang for the diagrams, Arash Ahmadian for the discussions.

References

Arash Ahmadian Aakanksha, Beyza Ermis, Seraphina Goldfarb-Tarrant, Julia Kreutzer, Marzieh Fadaee, Sara Hooker, and 1 others. 2024. The multilingual alignment prism: Aligning global and local preferences to reduce harm. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12027–12049.

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169.

Arash Ahmadian, Seraphina Goldfarb-Tarrant, Beyza Ermis, Marzieh Fadaee, Sara Hooker, and 1 others. 2024. Mix data or merge models? optimizing for diverse multi-task learning. *arXiv preprint arXiv:2410.10801*.

Afra Amini, Tim Vieira, Elliott Ash, and Ryan Cotterell. 2025. Variational best-of-n alignment. In *The Thirteenth International Conference on Learning Representations*.

Amanda Bertsch, Alex Xie, Graham Neubig, and Matthew Gormley. 2023. It's MBR all the way down: Modern generation techniques through the lens of minimum Bayes risk. In *Proceedings of the Big Picture Workshop*, pages 108–122, Singapore. Association for Computational Linguistics.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. 2024. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*.

Jianhao Chen, Zishuo Xun, Bocheng Zhou, Han Qi, Qiaosheng Zhang, Yang Chen, Wei Hu, Yuzhong Qu, Wanli Ouyang, and Shuyue Hu. 2025. Do we truly need so many samples? multi-llm repeated sampling efficiently scale test-time compute. *arXiv preprint arXiv:2504.00762*.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2023. Breaking language barriers in multilingual mathematical reasoning: Insights and observations. *arXiv preprint arXiv:2310.20246*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Team Cohere, Arash Ahmadian, Marwan Ahmed, Jay Alammar, Yazeed Alnumay, Sophia Althammer, Arkady Arkhangorodsky, Viraat Aryabumi, Dennis Aumiller, Raphaël Avalos, and 1 others. 2025. Command a: An enterprise-ready large language model. *arXiv preprint arXiv:2504.00698*.

Pierre Colombo, Nuno Guerreiro, Ricardo Rei, Daan Van, Luisa Coheur, and André Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*.

Jonathan Cook, Tim Rocktäschel, Jakob Foerster, Dennis Aumiller, and Alex Wang. 2024. Ticking all the boxes: Generated checklists improve llm evaluation and generation. *arXiv preprint arXiv:2410.03608*.

John Dang, Arash Ahmadian, Kelly Marchisio, Julia Kreutzer, Ahmet Üstün, and Sara Hooker. 2024a. RLHF can speak many languages: Unlocking multilingual preference optimization for LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13134–13156, Miami, Florida, USA. Association for Computational Linguistics.

John Dang, Shivalika Singh, Daniel D'souza, Arash Ahmadian, Alejandro Salamanca, Madeline Smith, Aidan Peppin, Sungjin Hong, Manoj Govindassamy, Terrence Zhao, and 1 others. 2024b. Aya expanse: Combining research breakthroughs for a new multilingual frontier. *arXiv preprint arXiv:2412.04261*.

Saurabh Dash, Yiyang Nan, John Dang, Arash Ahmadian, Shivalika Singh, Madeline Smith, Bharat Venkitesh, Vlad Shmyhlo, Viraat Aryabumi, Walter Beller-Morales, Jeremy Pekmez, Jason Ozuzu, Pierre Richemond, Acyr Locatelli, Nick Frosst, Phil Blunsom, Aidan Gomez, Ivan Zhang, Marzieh Fadaee, and 6 others. 2025. Aya vision: Advancing the frontier of multilingual multimodality. *Preprint*, arXiv:2505.08751.

Daniel Deutsch, Eleftheria Briakou, Isaac Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, and 1 others. 2025. Wmt24++: Expanding the language coverage of wmt24 to 55 languages & dialects. *arXiv preprint arXiv:2502.12404*.

Weihua Du, Yiming Yang, and Sean Welleck. 2025. Optimizing temperature for language models with multi-sample inference. *arXiv preprint arXiv:2502.05234*.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4506–4520, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10978–10993, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Christian Federmann, Tom Kocmi, and Ying Xin. 2022. NTREX-128 – news test references for MT evaluation of 128 languages. In *Proceedings of the First Workshop on Scaling Up Multilingual Evaluation*, pages 21–24, Online. Association for Computational Linguistics.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1396–1412, Seattle, United States. Association for Computational Linguistics.

Markus Freitag, Behrooz Ghorbani, and Patrick Fernandes. 2023. Epsilon sampling rocks: Investigating sampling strategies for minimum Bayes risk decoding for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9198–9209, Singapore. Association for Computational Linguistics.

Ashim Gupta and Vivek Srikumar. 2025. Test-time scaling with repeated sampling improves multilingual text generation. *Preprint*, arXiv:2505.21941.

Srishti Gureja, Lester James V Miranda, Shayekh Bin Islam, Rishabh Maheshwary, Drishti Sharma, Gusti Winata, Nathan Lambert, Sebastian Ruder, Sara Hooker, and Marzieh Fadaee. 2024. M-rewardbench: Evaluating reward models in multilingual settings. *arXiv preprint arXiv:2410.15522*.

John Hewitt, Christopher Manning, and Percy Liang. 2022. Truncation sampling as language model desmoothing. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3414–3427, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration. In *International Conference on Learning Representations*.

Sara Hooker. 2024. On the limitations of compute thresholds as a governance strategy. *Preprint*, arXiv:2407.05694.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Dylan J Foster, and Akshay Krishnamurthy. 2025. Is best-of-n the best of them? coverage, scaling, and optimality in inference-time alignment. *arXiv preprint arXiv:2503.21878*.

Yuki Ichihara, Yuu Jinnai, Tetsuro Morimura, Kenshi Abe, Kaito Ariu, Mitsuki Sakamoto, and Eiji Uchibe. 2025. Evaluation of best-of-n sampling strategies for language model alignment. *Transactions on Machine Learning Research*.

Daphne Ippolito, Reno Kriz, João Sedoc, Maria Kustikova, and Chris Callison-Burch. 2019. Comparison of diverse decoding methods from conditional language models. In *Proceedings of the 57th Annual Meeting of the Association for Computational*

Linguistics, pages 3752–3762, Florence, Italy. Association for Computational Linguistics.

Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4334–4353, Miami, Florida, USA. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, and 1 others. 2024. Findings of the wmt24 general machine translation shared task: the llm era is here but mt is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1–46.

Geza Kovacs, Daniel Deutsch, and Markus Freitag. 2024. Mitigating metric bias in minimum Bayes risk decoding. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1063–1094, Miami, Florida, USA. Association for Computational Linguistics.

Julia Kreutzer, Eleftheria Briakou, Sweta Agrawal, Marzieh Fadaee, and Kocmi Tom. 2025. D\'ej\a vu: Multilingual llm evaluation through the lens of machine translation evaluation. *arXiv preprint arXiv:2504.11829*.

Shankar Kumar and William Byrne. 2002. Minimum bayes-risk word alignments of bilingual texts. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*, EMNLP '02, page 140–147, USA. Association for Computational Linguistics.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. Rewardbench: Evaluating reward models for language modeling. https://huggingface.co/spaces/allenai/reward-bench.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Tianhao Wu, Banghua Zhu, Joseph E Gonzalez, and Ion Stoica. 2024. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. *arXiv preprint arXiv:2406.11939*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-reward: Bag of tricks for reward modeling in llms. *arXiv preprint arXiv:2410.18451*.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. Rewardbench 2: Advancing reward model evaluation. *arXiv preprint arXiv:2506.01937*.

Xiaoyu Tan Minghao Yang, Chao Qu. 2024. Inf-orm-llama3.1-70b.

Nguyen Nhat Minh, Andrew Baker, Clement Neo, Allen G Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent LLM outputs. In *The Thirteenth International Conference on Learning Representations*.

Ayomide Odumakinde, Daniel D'souza, Pat Verga, Beyza Ermis, and Sara Hooker. 2024. Multilingual arbitrage: Optimizing data pools to accelerate multilingual progress. *arXiv preprint arXiv:2408.14960*.

Max Peeperkorn, Tom Kouwenhoven, Dan Brown, and Anna Jordanous. 2024. Is temperature the creativity parameter of large language models? In *International Conference on Computational Creativity*.

José Pombal, Dongkeun Yoon, Patrick Fernandes, Ian Wu, Seungone Kim, Ricardo Rei, Graham Neubig, and André FT Martins. 2025. M-prometheus: A suite of open multilingual llm judges. *arXiv preprint arXiv:2504.04953*.

Matthew Renze. 2024. The effect of sampling temperature on problem solving in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356, Miami, Florida, USA. Association for Computational Linguistics.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8601–8629, Miami, Florida, USA. Association for Computational Linguistics.

Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, and 1 others. 2022. Language models are multilingual chain-of-thought reasoners. *arXiv preprint arXiv:2210.03057*.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2025. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*.

Guijin Son, Jiwoo Hong, Hyunwoo Ko, and James Thorne. 2025. Linguistic generalizability of test-time scaling in mathematical reasoning. *arXiv preprint arXiv:2502.17407*.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. 2025. The good, the bad, and the greedy: Evaluation of LLMs should not ignore non-determinism. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4195–4206, Albuquerque, New Mexico. Association for Computational Linguistics.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the wmt15 metrics shared task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273.

Benedikt Stroebl, Sayash Kapoor, and Arvind Narayanan. 2024. Inference scaling $\{F\}$ laws: The limits of llm resampling with imperfect verifiers. *arXiv preprint arXiv:2411.17501*.

Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast best-of-n decoding via speculative rejection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Nandan Thakur, Jianmo Ni, Gustavo Hernandez Abrego, John Wieting, Jimmy Lin, and Daniel Cer. 2024. Leveraging LLMs for synthesizing training data across many languages in multilingual dense retrieval. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7699–7724, Mexico City, Mexico. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ahmet Üstün, Viraat Aryabumi, Zheng-Xin Yong, Wei-Yin Ko, Daniel D'souza, Gbemileke Onilude, Neel Bhandari, Shivalika Singh, Hui-Lee Ooi, Amr Kayid, and 1 others. 2024. Aya model: An instruction finetuned open-access multilingual language model. *arXiv preprint arXiv:2402.07827*.

Junlin Wang, Shang Zhu, Jon Saad-Falcon, Ben Athiwaratkun, Qingyang Wu, Jue Wang, Shuaiwen Leon Song, Ce Zhang, Bhuwan Dhingra, and James Zou. 2025. Think deep, think fast: Investigating efficiency of verifier-free inference-time-scaling methods. *Preprint*, arXiv:2504.14047.

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*.

Zhilin Wang, Yi Dong, Olivier Delalleau, Jiaqi Zeng, Gerald Shen, Daniel Egert, Jimmy J Zhang, Makesh Narsimhan Sreedhar, and Oleksii Kuchaiev. Helpsteer 2: Open-source dataset for training top-performing reward models. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. 2024. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*. Survey Certification.

Ian Wu, Patrick Fernandes, Amanda Bertsch, Seungone Kim, Sina Khoshfetrat Pakazad, and Graham Neubig. 2025. Better instruction-following through minimum bayes risk. In *The Thirteenth International Conference on Learning Representations*.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. 2024a. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Zhaofeng Wu, Ananth Balashankar, Yoon Kim, Jacob Eisenstein, and Ahmad Beirami. 2024b. Reuse your rewards: Reward model transfer for zero-shot cross-lingual alignment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 1332–1353, Miami, Florida, USA. Association for Computational Linguistics.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Wen Yang, Junhong Wu, Chen Wang, Chengqing Zong, and Jiajun Zhang. 2025b. Language imbalance driven rewarding for multilingual self-improving. In *The Thirteenth International Conference on Learning Representations*.

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

Dongkeun Yoon, Joel Jang, Sungdong Kim, Seungone Kim, Sheikh Shafayat, and Minjoon Seo. 2024. LangBridge: Multilingual reasoning without multilingual supervision. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7502–7522, Bangkok, Thailand. Association for Computational Linguistics.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024a. Generative verifiers: Reward modeling as next-token prediction. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025a. Generative verifiers: Reward modeling as next-token prediction. In *The Thirteenth International Conference on Learning Representations*.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. 2025b. What, how, where, and how well? a survey on test-time scaling in large language models. *arXiv preprint arXiv:2503.24235*.

Yuanchi Zhang, Yile Wang, Zijun Liu, Shuo Wang, Xiaolong Wang, Peng Li, Maosong Sun, and Yang Liu. 2024b. Enhancing multilingual capabilities of large language models through self-distillation from resource-rich languages. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11189–11204, Bangkok, Thailand. Association for Computational Linguistics.

Wenhao Zhu, Shujian Huang, Fei Yuan, Shuaijie She, Jiajun Chen, and Alexandra Birch. 2024. Question translation training for better multilingual reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8411–8423, Bangkok, Thailand. Association for Computational Linguistics.

A Detailed Experimental Setup

Models and Language Coverage For the multilingual generative model, we consider two 8B models from different families: Aya-Expanse-8B (Dang et al., 2024b) and Qwen3-8B (Yang et al., 2025a). *Aya-Expanse-8B* is an open-weights multilingual LLM supporting 23 languages. It employs a post-training recipe focused on open-ended generative tasks that includes supervised fine-tuning, multilingual preference tuning (Aakanksha et al., 2024; Dang et al., 2024a) and synthetic data optimization (Odumakinde et al., 2024), and model merging (Ahmadian et al., 2024) to achieve strong competitive performance on open-ended benchmarks for models of this size. *Qwen3-8B* is an open-source dense model from the Qwen3 model family, supporting up to 119 languages and dialects. It was post-trained through distillation from larger models in the family and is used here in its "non-thinking" mode without reasoning. We focus on a subset of 7 languages which are covered by both models: *English, French, German, Japanese, Russian, Simplified Chinese, Spanish* prioritized due to their inclusion in generative benchmarks of interest. For WMT, we translate from English into all other languages.

Multi-Task Evaluation As outlined in section 1, math and translation have attracted the majority of research into test-time scaling, while open-ended generation tasks have received less attention. It is harder to fit a single method or reward model to the diverse challenges that open-ended generations pose. Our goal here is to take a wider view, which means considering both open-ended tasks and tasks with underlying correctness. We use the benchmarks summarized in Table 6 to measure the following for our multilingual evaluations:

- **Open-ended generation quality** on m-ArenaHard (Dang et al., 2024b), a translated version of the English Arena-Hard-Auto v0.1(Li et al., 2024) prompts. We measure win rate % using GPT-4o (`gpt-4o-2024-05-13`) as a judge, which is the standard choice for this benchmark. We also used **m-ArenaHard-v2.0** as our main test-set for open-ended evaluation. To create the m-ArenaHard-v2.0 test set, we obtain the 750 prompts from Arena-Hard-v2.0[7] and use papluca/xlm-roberta-base-language-detection[8] to perform language identification. Of these, the 498 identified as "English" were then translated into target languages by using an in-house state-of-the-art translation model.

- **Mathematical reasoning** For development, we obtain the GSM8K Parallel Translated Corpus (Chen et al., 2023) and group them by the original prompt. We then randomly select 250 prompt groups and select the same for all languages to avoid cross-lingual contamination. For each math problem, the model is instructed in the specific language to solve step-by-step and provide a final answer. Final answers are extracted from the step-by-step solutions and evaluated for accuracy using exact match to the correct answer, following `simple-evals` [9]. We test on MGSM (Shi et al., 2022), a manually translated version of 250 grade-school math problems from English GSM8K (Cobbe et al., 2021).

- **Translation quality** We use the development sets from WMT24 (Kocmi et al., 2024) for most language pairs with the except of `en-fr`, which we obtain from WMT15 (Stanojević et al., 2015) as our dev sets for machine translation and for the test set we used the WMT24++ dataset (Deutsch et al., 2025).

Model Serving: We use vLLM (Kwon et al., 2023) to generate outputs from our 8B models (AYA and QWEN), loading them with FP8 quantization and a maximum sequence length of 8,192 tokens. For the larger models (Command A and Gemini 2.0 Flash), we use their dedicated hosted APIs. For greedy decoding, we set top-k to 1, while for multi-sample generation we obtain five completions at specified temperature and min-p values.

B Choosing Judge and Reward Model

Table 7 compares Multilingual Reward-Bench (Gureja et al., 2024) scores for multiple generative multilingual LLMs. We add benchmark scores for Command A by running the official code released with the benchmark.[10] Remaining scores are taken from prior reports (Gureja et al.,

[7] https://github.com/lmarena/arena-hard-auto/blob/main/data/arena-hard-v2.0/

[8] https://huggingface.co/papluca/xlm-roberta-base-language-detection

[9] https://github.com/openai/simple-evals/tree/main

[10] https://github.com/Cohere-Labs-Community/m-rewardbench, commit `5e5a0d3` .

Name	Data Splits (# Prompts per Language)			Metric
	dev	devtest	test	
Arena	m-ArenaHard (250/250)		m-ArenaHard-v2.0 (498)	Win rate
MGSM	GSM8K-instruct-parallel (250/250)		MGSM (258)	Accuracy
WMT	WMT24/15 dev (997/1.5k)	NTREX (1997)	WMT24++ (960)	XComet-XL

Table 6: Overview of the **benchmarks** used in this work for open-ended generation (Arena), mathematical reasoning (MGSM), and machine translation (WMT). We compile `dev` and `devtest` splits to prevent overfitting our sampling and selecting methods to the `test` set. For WMT dev, French prompts were retrieved from WMT15 dev, the remaining ones from WMT24 dev. For WMT24++ with originally 998 instances, we skip those marked as "bad source".

Model	**Avg**
GPT-4o[1] (`gpt-4o-2024-08-06`)	81.1
GPT-4o[2] (`gpt-4o-2024-11-20`)	85.8
Aya Expanse 8B[1]	65.2
Llama 3.1 70B[1]	75.5
Gemma 2 9B[1]	76.6
M-Prometheus-14B[2]	79.5
Qwen2.5-14B-Instruct[2]	80.8
Command A (111B)	84.5

Table 7: Average accuracy on the M-RewardBench across 24 languages, comparing open models of various sizes (below) to GPT-4o variants. [1]: Results are copied from (Gureja et al., 2024), [2]: from (Pombal et al., 2025).

2024; Pombal et al., 2025). Table 8 details the performance for Command A for each language. According to this benchmark and the scores reported in (Gureja et al., 2024) and (Pombal et al., 2025), Command A is the best open judge, scoring closely to GPT-4o (and even outperforming an older variant).

There is further support in experiments by Kreutzer et al. (2025) where its agreement with pairwise human preferences from Chatbot Arena battles in multiple languages is close to GPT-4o's, with particular strengths in Chinese, Vietnamese, French, Turkish and Dutch.

On the English RewardBench benchmark (Lambert et al., 2024), classifier RMs are outperforming generative ones, so we pick the top-performing open model as our RM for BoN, `INF-ORM-Llama3.1-70B` (Minghao Yang, 2024), which is based on the multilingual Llama3.1-70B that supports English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai—which we suspect yields the strong crosslingual generalization.

Language	**Accuracy**
arb_Arab	84.61
ces_Latn	83.83
deu_Latn	85.08
ell_Grek	84.04
fra_Latn	85.36
heb_Hebr	83.62
hin_Deva	83.74
ind_Latn	84.35
ita_Latn	85.90
jpn_Jpan	84.94
kor_Hang	83.43
nld_Latn	86.32
pes_Arab	81.98
pol_Latn	84.91
por_Latn	86.30
ron_Latn	83.22
rus_Cyrl	84.12
spa_Latn	86.55
tur_Latn	82.96
ukr_Cyrl	83.83
vie_Latn	84.49
zho_Hans	84.59
zho_Hant	84.25
Avg	84.45

Table 8: Language breakdown of M-RewardBench scores for Command A.

C Temperature Sensitivity

Figure 5, fig. 6 and fig. 7 show how the score of *best*, *mean* and *worst* samples as we increase the temperature for m-ArenaHard, MGSM, and WMT, respectively and we see consistent trends of HOPE and RISK.

In Table 9 we report the significance across different sampling methods and we find that, for the majority of languages, the standard error is low with significant 95% CIs.

Finally, Table 10 presents a breakdown of how frequently the greedy sample is selected by the Judge MBR method within the hedge sampling setup. We find that the greedy sample is chosen approximately 35% of the time from a pool of five samples, which significantly exceeds random chance (20%) and demonstrate that the our hedging intervention contributes meaningfully to the overall quality of the samples pool.

D Selection Prompts

Listing 1 reports the judge prompt for CHOPS, listing 2 and listing 3 show the prompt to generate cross-lingual sample and the prompt to judge these samples for X-MBR respectively.

E Ablations

E.1 Choosing Single Temperatures

One could tune the single temperature, but in practice, resources invested in such tuning might have limited return. For our tasks and models, we measure the maximum HOPE at $\tau = 0.7$ of 48% but it is close to the value at $\tau = 0.8$ and $\tau = 0.9$ of 47% and 45%, thus the effects of choosing one over the other might be negligible. However, in Table 11 we can that compare to the lower temperature of $\tau = 0.3$ we see significantly higher hope at $\tau = 0.7$.

E.2 Token-level hedging with min-p sampling

We consider min-p sampling as an additional token-level hedging mechanism during generation. Figure 8 demonstrates that adding min-p consistently improves performance across selection methods compared to only Hedged Sampling. For multilingual win-rates evaluation, min-p provides substantial improvements for both RM BoN and CHOPS, while machine translation tasks show more modest but consistent benefits.

E.3 One-Pass Selection

Figure 9 compares our one-pass, checklisted selection method (CHOPS) against a simpler one-pass selection (OPS) setup without any grounding checklist. We plot the average win rates over greedy decoding for English •and non-English ◎languages. OPS achieves a high win rate in •(+9.0%) but under-performs in ◎(+5.3%), whereas CHOPS gives a more balanced outcome of +6.8% (•) and +7.1% (◎) win-rates over greedy. Notably the checklisted is less in tasks like MGSM where comparison criteria are more explicitly defined, Figure 10.

E.4 Sample Size Scaling

In fig. 11 we compare reward scores for mArena-Hard when sampling ($t = 0.7$, hedged) from Aya Expanse 8B beyond only five samples. When reward and selection metric are perfectly aligned, as in Figure 11, there is smooth improvement up to $N = 40$ for all languages.

However, we are working with imperfect selection methods that are not perfectly aligned with the evaluation metric, so we do not expect these to transfer to realistic use cases Huang et al. (2025); Stroebl et al. (2024). Figure 12 illustrates this issue, Win rate improvements over greedy are not developing smoothly across languages, and sometimes even losing to greedy. We observe that RM BoN is most reliable in its improvement with more samples, likely because it evaluates all samples in isolation and is thereby less affected by scale artifacts that limits pairwise (MBR) or direct (CHOPS) comparisons.

Finally, Figure 13 compares CHOPS and RM-BoN across 1-20 samples. CHOPS outperforms RM-BoN at smaller scales (5-15 samples) but falls behind at larger sample sizes. This indicates that CHOPS is more sample-efficient at lower scales, though its gains diminish at higher scales compared to RM-BoN.

F Evaluation Results

Table 13 shows the breakdown of baseline comparisons on the development set for hedged sampling ($\tau = 0.7$, $N = 5$).

Table 12 contains the breakdown into individual languages and tasks for the test set evaluations of hedged sampling ($\tau = 0.7$, $N = 5$).

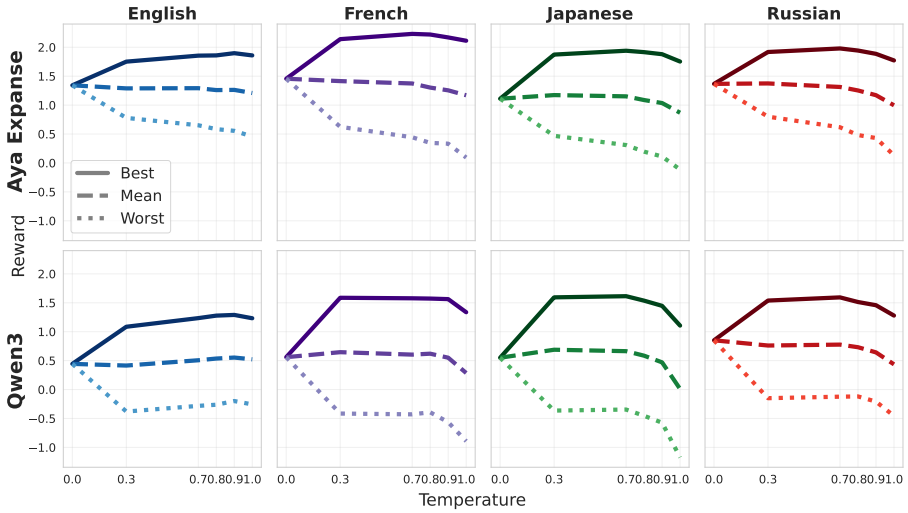


Figure 5: **Arena:** Evaluation score under different temperatures with $N = 5$ samples.

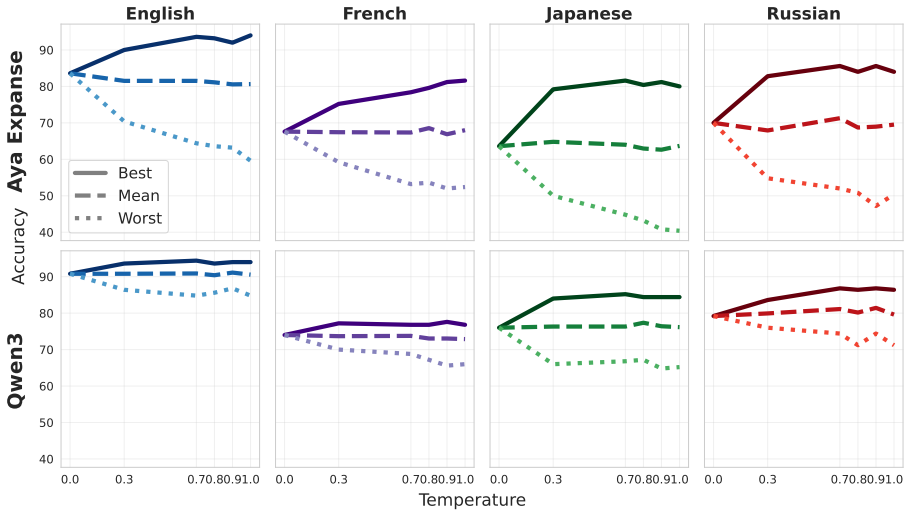


Figure 6: **MGSM:** Evaluation score under different temperatures with $N = 5$ samples.

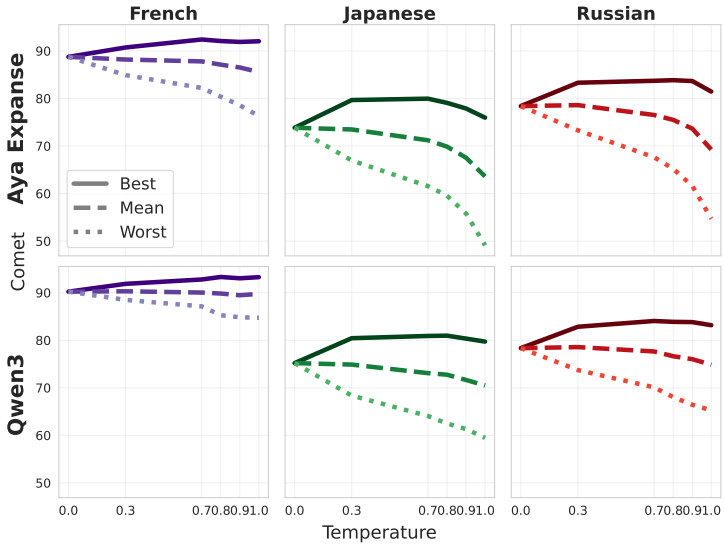


Figure 7: **WMT** Evaluation score under different temperatures with $N = 5$ samples.

```
"""
Please act as a fair judge. Based on the provided Instruction and Generated Texts in
    {language}, analyse the Generated Texts step by step and reply with the index
    of the best response.

(1) As a first step, write a tailored **evaluation checklist for the Instruction**
    that was given to an AI assistant.
This evaluation checklist should be a list of questions that ask whether or not
    specific criteria relevant to the Instruction were met by an AI assistants
    response.
Each question in the checklist should measure one distinct aspect of quality.
Criteria covered by your checklist could be explicitly stated in the Instruction, or
    be generally sensible criteria for the problem domain.
Factuality is always of high importance, but where relevant, the checklist should
    also take language and culture-specific context into account.
Questions should be concise and the checklist should not include unnecessary entries
    . Avoid vague wording and instead evaluate specific aspects of a response.

(2) As a second step, **compare the Generated Texts** with respect to the checklist.
    Write a brief explanation of the evaluation.

(3) In the third step, **output the index of the best Generated Text** according to
    the checklist evaluation.

## Output Format
Checklist: (each question should appear on a new line)
Q1: xxx
Q2: xxx

Explanation: xxx

Answer: [[ INDEX ]] (this should be an integer and nothing else; the index should be
    enclosed in double brackets as indicated)

## Evaluation Information

**Instruction**
{message}


**Generated Texts**
{generations}


Please analyse the Generated Texts according to a custom checklist and provide your
    selected text according to the guidelines. Remember to stick to the requested
    Output Format, providing the checklist questions and a short explanation and
    return the index of your selection inside double brackets [[]].
"""
```

Listing 1: Prompt Used for Checklisted One Pass Selection (CHOPS)

```
"""
Respond to the following instruction in {language_name}. Only return the answer in {
    language_name}.

{prompt}

Now Give your response  following the above instruction ONLY in {language_name}.
"""
```

Listing 2: Prompt Used for Cross-Lingual MBR (X-MBR)

Model	Sampling	Language	Std Error	95% CI
AYA	Hedged Random	Chinese	0.055	0.20 - -0.01
		English	0.055	0.20 - -0.02
		French	0.054	0.21 - -0.00
		German	0.054	0.21 - -0.01
		Japanese	0.054	0.18 - -0.03
		Russian	0.055	0.15 - -0.06
		Spanish	0.052	0.16 - -0.04
	Hedged	Chinese	0.058	0.15 - -0.07
		English	0.058	0.23 - -0.00
		French	0.057	0.13 - -0.10
		German	0.058	0.18 - -0.05
		Japanese	0.056	0.32 - 0.10
		Russian	0.057	0.22 - 0.00
		Spanish	0.056	0.16 - -0.06
	Random	Chinese	0.057	0.28 - 0.06
		English	0.057	0.19 - -0.04
		French	0.058	0.19 - -0.03
		German	0.058	0.19 - -0.04
		Japanese	0.057	0.28 - 0.05
		Russian	0.057	0.25 - 0.03
		Spanish	0.057	0.10 - -0.12
	Single	Chinese	0.062	0.23 - -0.01
		English	0.062	0.13 - -0.11
		French	0.063	0.17 - -0.08
		German	0.063	0.16 - -0.09
		Japanese	0.063	0.24 - -0.01
		Russian	0.062	0.19 - -0.05
		Spanish	0.062	0.17 - -0.08
QWEN	Hedged Random	Chinese	0.060	0.25 - 0.02
		English	0.061	0.23 - -0.01
		French	0.061	0.11 - -0.13
		German	0.061	0.23 - -0.01
		Japanese	0.061	0.13 - -0.11
		Russian	0.061	0.17 - -0.07
		Spanish	0.060	0.07 - -0.16
	Hedged	Chinese	0.059	0.19 - -0.04
		English	0.059	0.15 - -0.08
		French	0.058	0.15 - -0.08
		German	0.058	0.17 - -0.06
		Japanese	0.057	0.15 - -0.07
		Russian	0.058	0.24 - 0.01
		Spanish	0.057	0.28 - 0.06
	Random	Chinese	0.062	0.15 - -0.09
		English	0.061	0.31 - 0.08
		French	0.061	0.27 - 0.03
		German	0.061	0.21 - -0.03
		Japanese	0.062	0.15 - -0.09
		Russian	0.061	0.09 - -0.15
		Spanish	0.061	0.19 - -0.05
	Single	Chinese	0.063	0.15 - -0.10
		English	0.062	0.21 - -0.03
		French	0.063	0.24 - -0.00
		German	0.062	0.24 - -0.01
		Japanese	0.063	0.16 - -0.09
		Russian	0.063	0.18 - -0.06
		Spanish	0.063	0.12 - -0.13

Table 9: Comparison of the significance of different temperature sampling methods across models and languages using standard errors and 95% confidence intervals of the win-rates on m-ArenaHard with Judge MBR selection.

Model	Chinese	English	French	German	Japanese	Russian	Spanish	Mean ± Std
Aya	36.0	37.6	36.8	34.8	34.4	32.8	38.4	35.8 ± 2.0
Qwen	39.2	36.8	35.2	32.0	30.8	35.2	36.4	35.1 ± 2.9

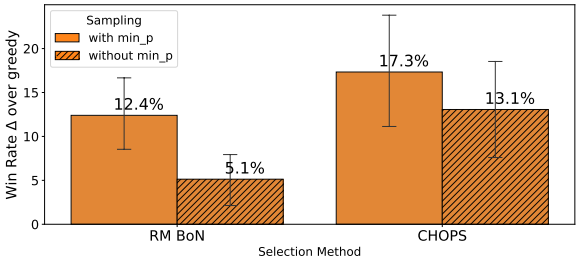
Table 10: Frequency of Greedy sample selection in Hedge sampling (m-ArenaHard)

```
"""
[Instruction]
Please act as an impartial multilingual judge and evaluate the quality of the
    response provided by an AI assistant to the user question displayed below. In
    addition to the user question, you are also given a reference answer that might
    be written in another language. This is the best possible answer provided by a
    human expert who might not speak the target language.
You should evaluate the assistants response based on this. A good assistants answer
    should share the content and style of the reference answer, but it does not have
     to match the language of the reference. Begin your evaluation by providing a
    short explanation. Be as objective as possible.
After providing your explanation, you must rate the response on a scale of 1 to 10
    by strictly following this format:
"[[rating]]", for example: "Rating: [[5]]".

[Question]
{question}

[Reference Answer]
{reference}

[The Start of Assistants Answer]
{candidate}
[The End of Assistants Answer]
"""
```

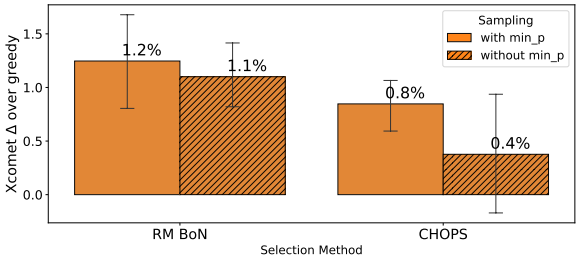
Listing 3: Prompt Used for Judging samples in X-MBR

Setting	Task	Hope	Risk
Temperature = 0.3			
English	Arena	30.60	-41.79
	MGSM	7.14	-16.67
	Average	**18.87**	**-29.23**
Non-English	Arena	49.66	-52.52
	MGSM	14.45	-17.54
	Average	**32.06**	**-35.03**
Temperature = 0.7			
English	Arena	38.81	-51.49
	MGSM	11.90	-23.81
	Average	**25.36**	**-37.65**
Non-English	Arena	55.51	-65.31
	MGSM	18.96	-23.70
	Average	**37.23**	**-44.50**

Table 11: Hope (best-case improvement) and risk (worst-case drop) of sampling at temperatures 0.3 and 0.7, relative to greedy decoding, across tasks and languages for Aya 8B. Values are reported as percentages.



(a) Multilingual win-rates.



(b) Machine translation

Figure 8: Effect of adding min-p sampling across different selection methods. Min-p provides consistent improvements across both methods and tasks. Results are on Arena (left) and WMT (right) dev splits, using AYA.

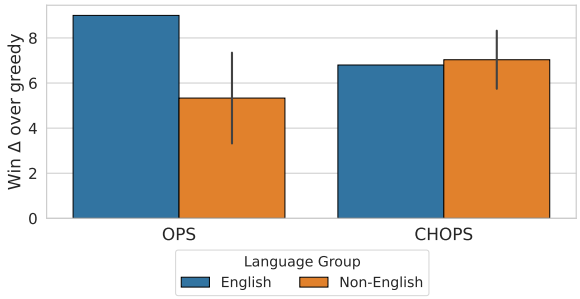


Figure 9: **Comparison of CHOPS** (with checklists) versus OPS (without checklists). Self-generated checklists perform better on average across languages.

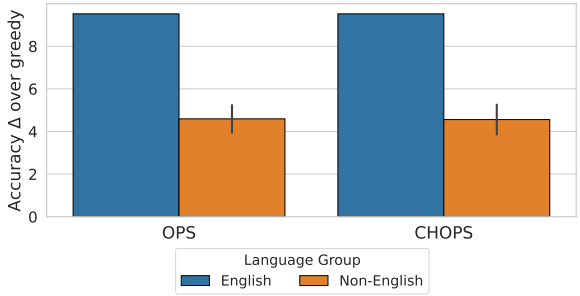


Figure 10: CHOPS vs OPS on MGSM: comparing the effect of check-listed selection on close ended tasks show negligible differences.

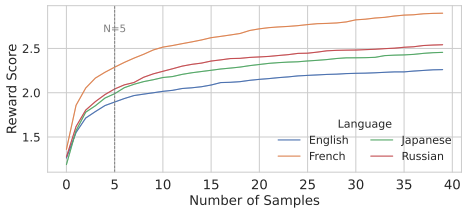


Figure 11: Reward Score Scaling as we increase the sample size from from 1 sample to 40 samples.

Model	Task	Language	CHOPS	Judge MBR	Reward BoN	X-MBR	Greedy
AYA	Arena	Chinese	18.40 ± 5.77	17.20 ± 5.79	7.60 ± 5.65	6.80 ± 5.70	–
		English	13.60 ± 5.63	6.40 ± 5.78	5.20 ± 5.49	13.20 ± 5.72	–
		French	13.20 ± 5.92	7.60 ± 5.68	8.40 ± 5.65	6.80 ± 5.68	–
		German	11.60 ± 5.75	5.20 ± 5.81	0.80 ± 5.59	16.40 ± 5.43	–
		Japanese	9.20 ± 5.83	4.00 ± 5.65	-0.40 ± 5.58	10.40 ± 5.63	–
		Russian	2.00 ± 5.72	-5.20 ± 5.67	8.80 ± 5.46	12.40 ± 5.72	–
		Spanish	24.00 ± 5.83	12.00 ± 5.62	5.60 ± 5.57	8.00 ± 5.72	–
	MGSM	Chinese	8.56 ± 2.59	7.36 ± 2.59	13.96 ± 2.46	12.16 ± 2.59	68.64
		English	7.76 ± 2.03	7.36 ± 2.03	13.16 ± 2.30	9.36 ± 1.93	77.84
		French	9.04 ± 2.79	7.84 ± 2.87	13.84 ± 2.88	10.24 ± 2.82	62.96
		German	5.12 ± 2.72	5.92 ± 2.71	10.32 ± 2.64	6.72 ± 2.63	73.68
		Japanese	6.56 ± 2.75	6.16 ± 2.86	10.36 ± 2.89	9.36 ± 2.75	68.64
		Russian	6.08 ± 2.64	5.68 ± 2.71	11.68 ± 2.53	8.48 ± 2.50	73.12
		Spanish	8.96 ± 2.41	8.56 ± 2.35	12.96 ± 2.22	9.76 ± 2.30	70.24
	WMT	Chinese	0.53*	0.10	2.59	-1.05	76.09
		French	0.39	0.09	2.55*	-2.14***	79.91
		German	0.17	0.04	0.86	-1.05***	91.88
		Japanese	1.12	0.65	3.03	-3.62***	78.99
		Russian	0.60*	0.31	2.86*	-2.73***	82.51
		Spanish	0.28**	0.01	1.68***	-0.74*	87.90
QWEN	Arena	Chinese	0.80 ± 6.13	-4.00 ± 5.94	8.80 ± 5.71	13.60 ± 6.01	–
		English	7.20 ± 6.10	4.80 ± 5.90	12.80 ± 5.58	9.60 ± 5.88	–
		French	8.00 ± 6.08	1.20 ± 5.77	8.40 ± 5.60	18.40 ± 5.79	–
		German	8.80 ± 5.96	-2.40 ± 5.83	-0.40 ± 5.71	12.00 ± 5.90	–
		Japanese	8.80 ± 6.10	6.40 ± 5.65	8.80 ± 5.82	4.40 ± 5.72	–
		Russian	8.00 ± 6.06	2.40 ± 5.84	3.60 ± 5.62	12.00 ± 5.85	–
		Spanish	8.00 ± 5.98	4.80 ± 5.73	16.00 ± 5.65	4.80 ± 5.82	–
	MGSM	Chinese	3.04 ± 2.41	3.04 ± 2.3	6.44 ± 2.22	4.24 ± 2.25	83.76
		English	-0.88 ± 1.72	-0.88 ± 1.80	2.12 ± 1.60	-0.08 ± 1.72	95.68
		French	2.48 ± 2.74	2.08 ± 2.74	4.28 ± 2.72	2.88 ± 2.72	79.92
		German	2.56 ± 2.48	2.16 ± 2.55	4.56 ± 2.50	4.96 ± 2.50	87.04
		Japanese	3.28 ± 2.59	3.68 ± 2.61	6.48 ± 2.50	4.88 ± 2.61	80.72
		Russian	1.36 ± 2.3	2.56 ± 2.41	3.96 ± 2.25	2.96 ± 2.32	89.04
		Spanish	2.48 ± 2.15	2.48 ± 2.17	5.28 ± 2.20	4.48 ± 2.15	86.72
	WMT	Chinese	0.51**	-0.29	2.18**	-0.23	78.42
		French	1.75*	0.06	2.81***	0.09**	77.71
		German	0.87**	-0.43***	1.56***	-0.23	89.86
		Japanese	1.69	-0.68	3.75*	-0.60	74.36
		Russian	2.21***	-1.68*	3.86***	-1.26	77.64
		Spanish	1.21**	-0.36**	2.24***	0.26	85.96

Table 12: Breakdown of test set results: Judge based Methods with Hedge sampling with $\tau = 0.7$. WMT Significance: * p < 0.05, ** p < 0.01, *** p < 0.001. MGSM std-err are reported for the selection methods absolute score (not the difference).

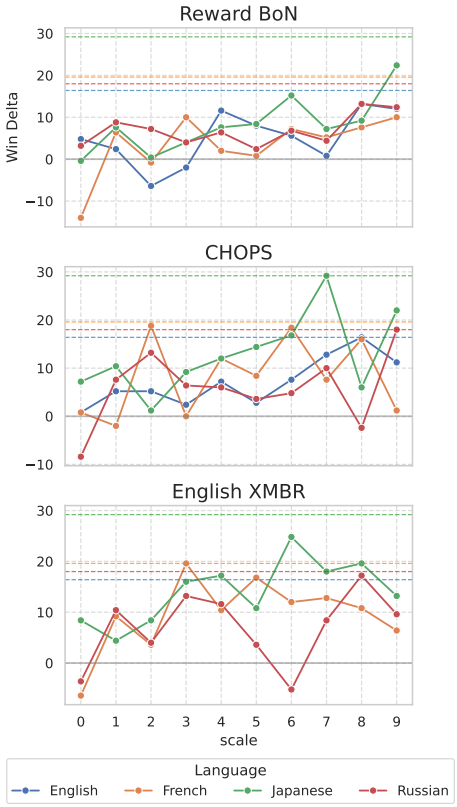


Figure 12: **Scaling pool sample size** using $\tau = 0.7$ hedged sampling for selected languages with different selection methods on mArenaHard dev set.

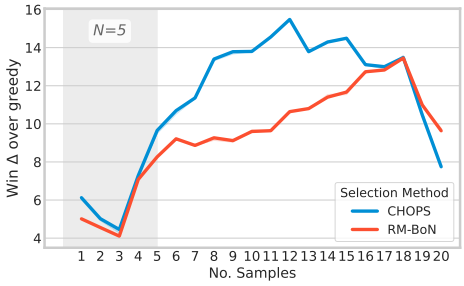


Figure 13: Win-rate gains from CHOPS and RM-BoN with increasing sample size (1-20) across all non-English languages for AYA on m-ArenaHard.

Model	Task	Language	Likelihood	Sim. MBR	BoN	Judge MBR
Aya	m-ArenaHard	Chinese	4.00	-6.40	5.20	4.00
		English	-7.60	10.40	12.00	11.20
		French	-15.20	0.40	14.80	1.60
		German	-12.80	-1.20	17.20	6.40
		Japanese	0.80	-1.20	16.40	20.80
		Russian	-6.80	2.80	11.60	11.20
		Spanish	-6.00	3.60	4.00	4.80
	MGSM	Chinese	1.76	2.96	12.16	9.36
		English	13.92	16.72	13.92	17.92
		French	2.16	2.56	4.56	4.96
		German	0.80	2.00	6.40	4.80
		Japanese	-0.32	4.88	7.68	8.88
		Russian	-0.08	0.72	7.52	3.52
		Spanish	-0.08	1.52	6.72	4.72
	WMT	Chinese	3.28	1.06	0.92	-0.07
		French	0.61	-0.07	1.04	-0.09
		German	0.42	0.39	0.32	0.04
		Japanese	-0.19	0.10	-0.18	0.11
		Russian	0.43	-0.07	1.08	0.15
		Spanish	-0.08	-0.42	1.49	0.08
Qwen	m-ArenaHard	Chinese	-1.60	-2.40	3.20	7.20
		English	-7.20	0.80	0.80	3.20
		French	1.60	-1.20	1.60	3.20
		German	-7.60	7.20	-4.00	5.20
		Japanese	-6.00	-1.20	7.20	4.00
		Russian	-1.60	-2.40	-1.60	12.40
		Spanish	-4.00	-1.20	8.80	16.80
	MGSM	Chinese	-1.44	0.96	3.36	2.16
		English	-0.88	-0.48	2.32	0.32
		French	-1.20	0.40	2.00	1.60
		German	0.72	1.12	2.72	1.52
		Japanese	1.12	1.12	5.12	2.72
		Russian	1.28	0.48	4.48	1.68
		Spanish	0.00	-0.40	0.40	0.80
	WMT	Chinese	0.26	-0.08	1.28	0.53
		French	0.14	-0.28	1.17	0.52
		German	0.80	0.27	1.99	1.37
		Japanese	0.51	0.17	1.941	1.61
		Russian	0.68	0.26	2.27	1.11
		Spanish	0.50	0.09	1.37	0.68

Table 13: Baseline vs. Judge MBR: breakdown by model, task, and language.