Ladders in Chaos: When, How, (and Perhaps Why) Does Test-Time Scaling Improve LLM Machine Translation

Anonymous ACL submission

Abstract

Two forms of test-time scaling for Large Language Models (LLMs) have emerged as effective and widely adopted paradigms: **sequential**, in which later answer attempts depend on earlier ones, and **parallel**, such as i.i.d. sampling with reranking. In this study, we investigate their properties in translation. First, our study shows that sequential sampling provides a more diverse and effective pool of samples for smaller budgets. Second, we interrogate the nature of test-time scaling through a multidimensional manual analysis. Human analysis of the Best-of-N translations demonstrates that sequential sampling substantially improves translation fluency and naturalness, but degrades accuracy, particularly under large inference budgets. Finally, we suggest an explanation of the mechanism through which sequential scaling improves machine translation. Our controlled analysis partially attributes the success of sequential self-improvement to the model's access to a larger target-side context. Ablation experiments on sequential sampling demonstrate its robustness across different sampling temperatures, while also revealing sensitivity to context construction, suggesting directions for future improvement.

1 Introduction

Large Language Model (LLM) based translation systems represent the current state of the art in translation quality (Kocmi et al., 2024, 2025). Moreover, LLMs have also been found to be highly fertile ground for further improvement through test-time scaling approaches (Snell et al., 2024; Muennighoff et al., 2025; Wu et al., 2025a; Tan et al., 2025, 2026). To understand the potential and effect of test-time scaling for machine translation, we study two commonly used paradigms: (1) **parallel**, for which the translations are *sampled* independently, followed by reranking using external verifiers, *i.e.*, Best-of-N (BoN) sampling, which in the machine translation community is known as Quality-Aware Decoding (QAD, Fernandes et al., 2022); and (2) **sequential**, where translations are revisited during subsequent rounds of generation *in a self-improving manner*. This paradigm has demonstrated effectiveness and has been interpreted as an advanced reasoning strategy, namely self-reflection (Madaan et al., 2023b; Guo et al., 2025; Muennighoff et al., 2025).

Although both sequential and parallel sampling explore more of the decoding space at the cost of increased computation, their properties and differences in machine translation have not been thoroughly studied: (1) which sampling strategy is more sample-effective and when; (2) in which dimensions of translation evaluation would parallel and sequential be different? (3) what mechanism drives this difference; and (4) what is the effect of context shaping on sequential sampling.

The outcome of our analyses can be summarized as follows:

- Sequential sampling explores the output spaces more effectively, compared to parallel sampling, especially under a limited budget.
- According to human evaluation results, sequential sampling improves fluency, accuracy and naturalness of translations; however, relative to parallel sampling, sequential can on average lose in accuracy.
- Our controlled experiments attribute the gains of sequential sampling in part to its access to a larger target-side context, rather than to the more commonly assumed self-reflection capability (Madaan et al., 2023b; Muennighoff et al., 2025).
- We further show that simply controlling the context in sequential sampling yields clear performance gains across languages and evaluation metrics.

2 Background

Test-time scaling. This method uses additional inference-time compute to improve model performance. Sequential test-time scaling in LLM research is typically realized through a combination of Chain-of-Thought (CoT, Wei et al., 2022) and iterative multi-turn generation. The former is generally regarded as a manifestation of reasoning capability (Guo et al., 2025), whereas the latter is often referred to as self-refinement (Madaan et al., 2023a). OpenAI's o1 model (Jaech et al., 2024) demonstrated such capabilities, while Muennighoff et al. (2025) extended the inference budget by force-generating a special "Wait" token, when a model tries to end generation.

Although promising, the underlying mechanisms behind this paradigm remain unclear, and its effectiveness varies considerably across domains. For example, Sprague et al. (2025) show that CoT-style reasoning is primarily effective for math and symbolic reasoning tasks, while yielding substantially smaller gains on other types of tasks. In translation, Wu et al. (2025a) demonstrate that explicitly decomposing the translation process through CoT does not lead to clear performance improvements. Instead, simply prompting LLMs to "translate again for a better version" yields even better results. These findings motivate our research to focus on iterative self-refinement as the main form of sequential test-time scaling.

Self-refinement. The refinement discussion for machine translation can be traced back to the automatic post-editing (APE) line of work, which has a long history in statistical machine translation (Knight and Chander, 1994; Shen et al., 2004; Simard et al., 2007; Correia and Martins, 2019; Fernandes et al., 2022). Recent refinement approaches investigate passing feedback directly to LLMs to steer subsequent generation. These methods can generally be divided into two categories based on the source of feedback: (1) approaches that leverage external translation quality signals (Huang et al., 2024; Wang et al., 2025a), and (2) approaches that generate feedback internally (Ki and Carpuat, 2024; Feng et al., 2025), a process commonly referred to as self-reflection or self-refinement (Madaan et al., 2023a).

In our study, we follow (Wu et al., 2025a) and consider a minimal setting for self-refinement to emphasize simplicity (§3.2). It is worth noting that although we employ an external metric model for Best-of-N selection in this study, this is intended to demonstrate the potential of the generated sample pool; the metric signal is never used to guide the model's behavior.

Sequential and parallel sampling. Our work builds on top of the somewhat tangential line of work which targets sampling strategies for diverse LLM outputs. Parallel (i.i.d.) or ancestral sampling techniques were developed and improved by the LLM community to trade-off between creativity and precision (Holtzman et al., 2020; Basu et al., 2021; Hewitt et al., 2022; Minh et al., 2025; Vilnis et al., 2023). Broadly, parallel sampling and its variants are used for Quality-Aware Decoding (Fernandes et al., 2022) or Minimal Bayes Risk decoding (Kumar and Byrne, 2004; Eikema and Aziz, 2020); to enable test-time controllability (Mudgal et al., 2024; Deng and Raffel, 2023); conformal generative modeling (Kladny et al., 2025), reasoning with diverse decoding paths (Wang et al., 2024); ambiguity resolution (Kobalczyk et al., 2025; Chen et al., 2025; Saparina and Lapata, 2025).

Sequential (non-i.i.d.) sampling in its various forms is often used as a way to improve sampling efficiency. Ilia and Aziz (2024) prompt API models to enumerate responses; Saparina and Lapata (2024) prompt language models to enumerate possible interpretations of ambiguous questions; Troshin et al. (2025b); Zhang et al. (2025) formulate variants of sequential sampling to generate diverse math solutions.

3 Sequential and Parallel Sampling

In this section, we formally define parallel and sequential sampling.

3.1 Parallel Sampling

Given a language model p_θ and a source text x to be translated, the output can be viewed as a sample $y \sim p_\theta(\cdot \mid I(x))$, where $I(x)$ is a prompt that includes x as a component to request translation. In contrast to beam-search, parallel sampling draws **translations** independently t samples from a LLM conditioned on the same input prompt:

$$y^{(i)} \sim p_\theta(\cdot|I(x)) \text{ for } i \in \{1, \ldots, t\}\,; \quad (1)$$

In Appendix A.1, we list the prompt format we used for our experiments. We do not use any form of prompt tuning, but we explicitly specify output format expectations.

3.2 Sequential Sampling

Sequential sampling for translation is a multi-turn process in which the **translations** of previous steps serve as context for the next prompt. For a turn t, *sequential* samples translations y as follows:

$$y^{(t)} \sim p_\theta \left(\cdot \mid I^{(t-1)}, y^{(t-1)}, ..., I^{(1)}, y^{(1)}, I(x) \right), \quad (2)$$

where $I^{(\cdot)}$ is a prompt requesting a model to revisit a translation to improve it (Wu et al., 2025a). We use the same prompt for each revisiting turn, *i.e.*, $I^{(\cdot)} := I$. The revisiting prompt I in our case is:

"Please translate again for a better version."

All sequential sampling strategies in this work use this prompt, because (1) it provides a maximally simplified setting, (2) it emphasizes the quality requirement, and (3) Wu et al. (2025a) show that this minimal prompt achieves the best results on the WMT24 dataset (Kocmi et al., 2024), outperforming carefully designed step-by-step prompting methods (Briakou et al., 2024).

Sequential sampling follows the test-time scaling framework in LLMs research: multiple rounds of translation are introduced at inference time, while explicitly requiring the model to produce better translations, analogous to test-time reflection ability in LLMs (Madaan et al., 2023b; Muennighoff et al., 2025). Notably, this setting removes the explicit chain-of-thought (CoT) process, as recent studies (Wu et al., 2025a; Rajaee et al., 2026) have shown that explicit CoT provides no clear benefit for general machine translation.

3.3 Sampling with a Selector

Two sampling strategies explore the decoding space in different ways. Our study investigates their respective properties in terms of randomness and potential of performance. However, investigating the performance scaling is nontrivial, as it requires identifying the best outputs from a large pool of stochastic generations. To this end, we follow a simple instance of Quality-Aware Decoding (QAD) strategy (Fernandes et al., 2022) and employ an external reference-free metric model (Rei et al., 2023) as the selector. This setting is often referred to as Best-of-N sampling in LLMs research.

More specifically, for a batch of translation $y_p^{1...N}$ and $y_s^{1...N}$, generated by *parallel* and *sequential* sampling given the same input x, we investigate the diversity and performance under the following three settings:

1. **Parallel w/ selector**: we independently sample N outputs for each source sentence, then use a reference-free metric model to select and report the corresponding best performance.
2. **Sequential w/o selector**: we sequentially sample N output for each source sentence, then directly report average translation performance in each sampling step.
3. **Sequential w/ selector**: we sequentially sample N output for each source sentence, then use a reference-free metric model to select and report the corresponding best performance.

Figure 1 exemplifies the three decoding strategies. In this work, we vary the sampling budget for each source sentence from 1 to N across the three settings to illustrate the variation in properties and performances. In the following study, we sample under the temperature of 1.0, except for temperature ablations (see Appendix H.2). We discuss other generation details in Appendix B.

4 Experimental Setup

4.1 Dataset

We use the WMT24++ dataset (Deutsch et al., 2025),[1] intended for evaluation of MT systems, as our test set, because 1) it covers a variety of domains and 2) the translation references in WMT24++ are human-written and subsequently post-edited by professional translators, ensuring the highest possible data quality. We filter out sentence pairs that are marked as low-quality or invalid using 'is_bad_source' labeled field, following WMT24++ recommendations.[2] To maximize the reliability of the metric models we used, we conduct experiments on a subset of 6 well-supported high– and medium-resource language directions, *i.e.*, `en`→`zh`,`de`,`ru`,`nl`,`ro`,`ar`,[3] representing different writing systems and language families.

4.2 Models

We experiment with capable open-source instruction-tuned models from the Qwen3 family across different parameter scales (Yang et al., 2025), namely `Qwen3-32B-Instruct` and `Qwen3-4B-Instruct-2507`; Following Wu et al. (2025a), we also conduct experiments with a commercial LLM `GPT-4o-mini`.

[1] Extended version of the WMT24 (Kocmi et al., 2024).
[2] This leaves 960 segments for each language.
[3] zh: `zh_CN`, ar: `ar_SA`

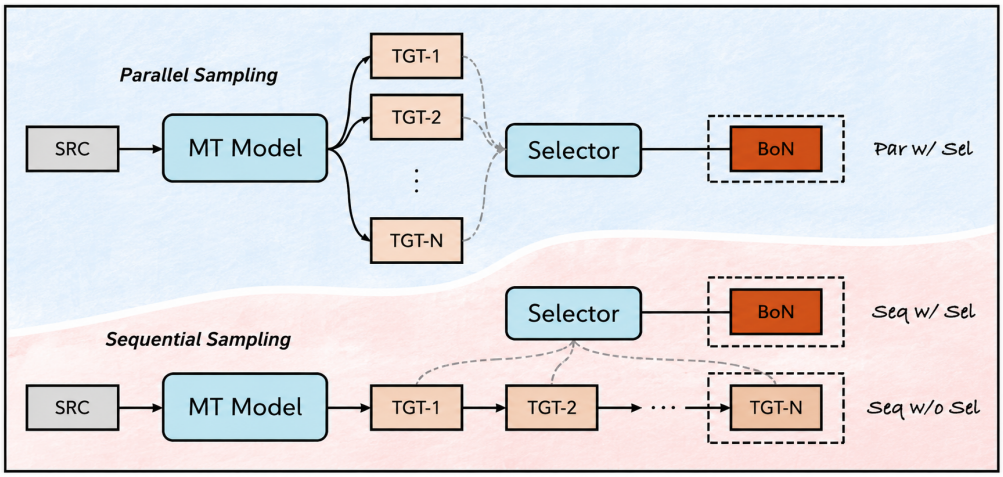


Figure 1: Illustration of the three sampling strategies investigated in this paper.

4.3 Metrics

Unless otherwise specified, we use CometKiwi-XL, the official reference-free metric for WMT24 (Kocmi et al., 2024) evaluation, as the selector for the following experiments. Meanwhile, to maximize the difference compared to the selector in case of excessive metric hacking (Skalse et al., 2022; Kovacs et al., 2024; Wu et al., 2025b), we report translation quality in a few widely used metric models, including a hybrid metric used in the QE and reference-based modes, namely MetricX-24-Hybrid-XXL (qe) and MetricX-24-Hybrid-XXL (ref); a reference-based COMET; a metric trained using a reward modeling objective (ReMedy-9B-22), used in the QE and reference-based mode. For the main results, we plot a translation metric of a selected sample (Best-of-N, or the last output of the self-refinement strategy) at a given budget level, where budget denotes the number of translation rounds.

5 An Overview of Sampling Effectiveness

In this section, we first give an overview of the effectiveness across different sampling strategies and evaluation metrics, under progressively increasing computation budget. Figure 2 presents the results for test-time scaling under multiple settings. As described in §4.3, we use `CometKiwi-XL` as a selector. We summarize the key findings as follows.

The "Translate Again" strategy works better in the early rounds. The green line in Figure 2 denotes the performance of "Translate Again" strategy in different metrics. It is clear that within the first few rounds (*e.g.*, 2 or 3), there are substantial and consistent performance improvements over the first round across all language directions and evaluation metrics. For example, for `en`→`ru` direction, the improvements reach approximately 1.5 COMET, 1.0 MetricX24-ref, and 1.5 MetricX24-qe points when employing the second round of translation. Our human evaluation aligns well with this finding (Figure 3). These findings are also aligned with Wu et al. (2025a), who report the improvements for early rounds. Our main finding patterns persist for commercial LLMs as well (see Figure 11 for the results with `GPT-4o-mini`).

Recent LLMs research attributes these improvements to LLMs' reflection ability (Madaan et al., 2023b; Muennighoff et al., 2025). In §6.2, we revisit this phenomenon and provide a more concrete explanation. Next, for later rounds of translation (*e.g.*, more than 5 rounds), the performance quickly plateaus and may even decline in some of the languages and metrics. In §6.3, we analyze the output space and partially attribute the plateau effect to the reduced marginal generation diversity.

Equipping selector is highly effective. The blue and orange lines in Figure 2 represent the results of parallel and sequential sampling with an external selector under different computation budgets, respectively. Notably, both methods substantially outperform direct translation (*i.e.*, a budget of 1), even under very limited budgets (*e.g.*, 2). Increasing the sampling size further improves performance across most metrics and language directions, with gains becoming substantial and beginning to converge for budget-30. The first column (in gray) in Figure 2 corresponds to the setting in which

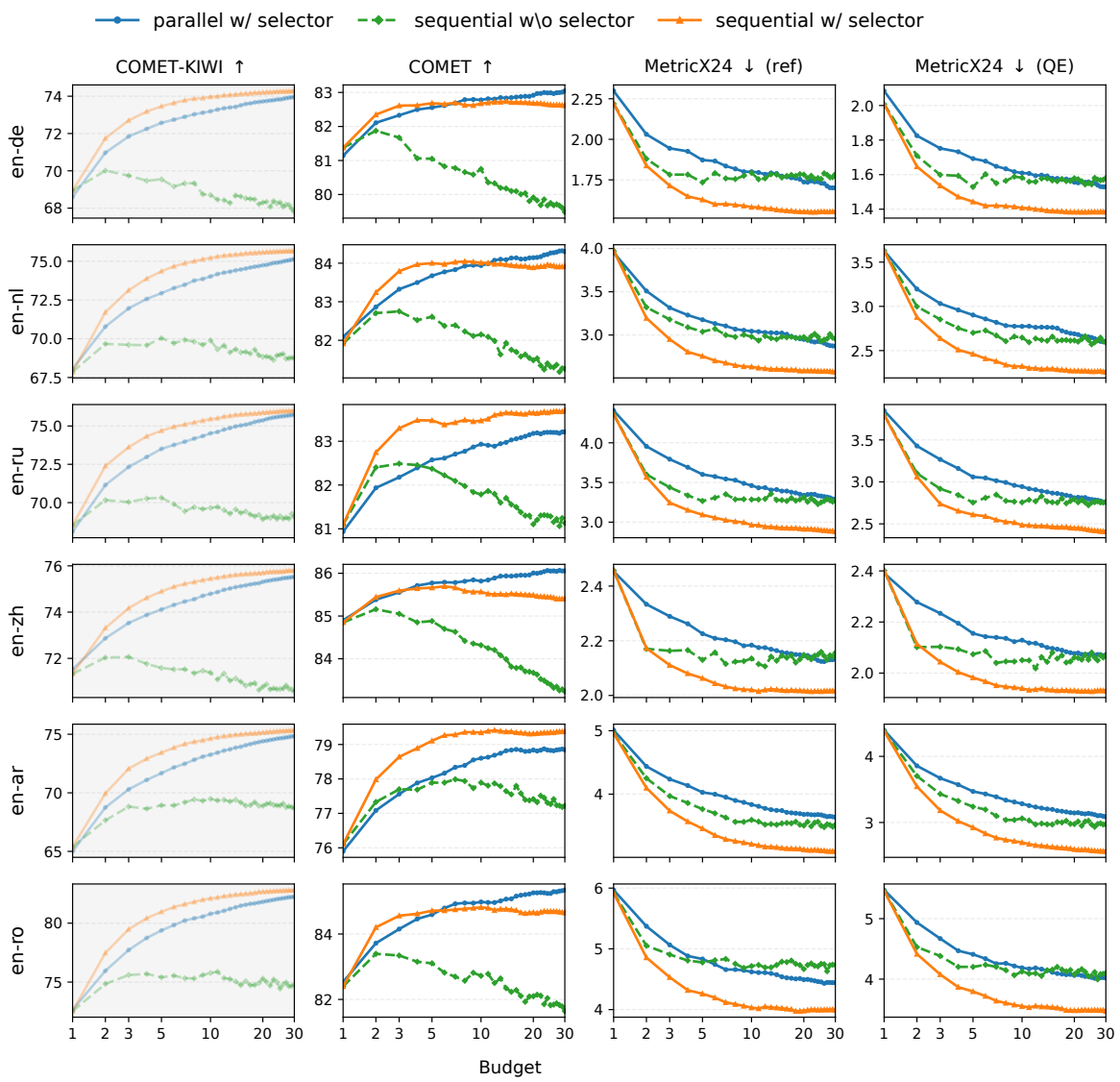


Figure 2: Quality vs. budget for the `Qwen-3-32B-Instruct` model. Budget denotes the number of rounds of translation for sequential or parallel. An additional evaluation with the ReMedy metric is shown in Figure 9.

CometKiwi-XL is used as both the selector and the evaluator,[4] shown here only to profile the Best-of-N sampling efficiency. Although the early iterations of "Translate Again" clearly improve average translation quality, the quality can degrade as the number of samples increases (at least according to COMET).[5]

By comparing results for sequential with and without selector (orange vs green), we observe that equipping selector allows us select better samples, given higher compute budgets. This suggests that the **language model alone cannot consistently produce better samples given more budget rounds**, and highlights the need for selector. We thus treat sequential sampling without selector more as a more stochastic and low-trust process.

Sequential sampling explores the decoding space more effectively, especially under relatively lower sampling budgets. Automatic evaluation results show that sequential sampling outperforms parallel sampling for smaller computational budgets (in all metrics), but sometimes loses to parallel for higher compute budgets (for the COMET metric); for the MetricX-24 metric, sequential always outperforms parallel. Overall results are very promising for sequential sampling, but slightly contradictory *w.r.t.* the performance ceiling, given the difference over metrics. To understand the results better, we conduct human annotation (§6.1) to inspect the improvements targeting not a single qual-

[4] A setting most prone to metric hacking.

[5] Our findings match the observation that COMET tends to emphasize adequacy over fluency (Park and Padó, 2024).

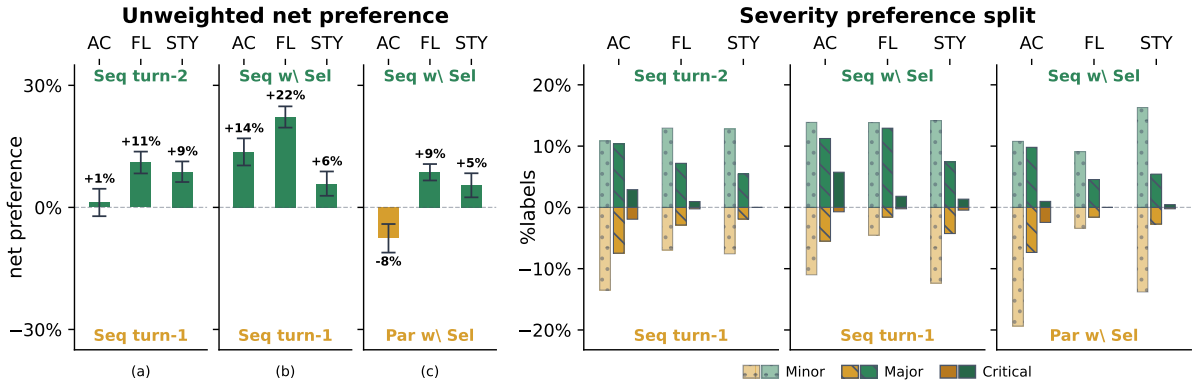


Figure 3: Averaged over 5 languages, annotation results for **average preference** using MQM dimensions: accuracy (AC), fluency (FL) and style (STY) for the `Qwen3-32B-Instruct` model. (left) shows the proportion of unweighted net preference (in %), and (right) presents preferences counts split by severity for each dimension. Results are presented for (i) sequential turn-1 vs turn-2, (ii) sequential with selector (budget-30) vs turn-1, (iii) sequential vs parallel, both with selector for budget-30. Error bars represent standard error of the mean (mean ± 1SE).

ity score, but 3 distinct MQM quality dimensions.

6 Analysis

6.1 Human evaluation

To better understand the automatic results for sequential vs parallel, we ask annotators to compare translations under three different settings: (a) sequential turn-1 vs turn-2, (b) sequential with selector vs turn-1, and (c) sequential vs parallel sampling with selector. We focus on three MQM dimensions (Freitag et al., 2021): accuracy (AC), fluency (FL) and style (STY), see Appendix C.1 for detailed annotation policy. Results are presented in Figure 3 and categorized into "Unweighted preference" (left) and "Severity split" (right) versions. The former figure represents the net ratio of winning translations (average) between a pair of compared methods, while the latter denotes the proportion of annotations for each dimension split by their error severity (minor, major, critical), where green or orange marks the direction (win/lose).

Compared with direct translation, sequential sampling yields clear improvements across all dimensions with just one additional round in Figure 3 left-(a). This performance gap is further amplified when combined with a selector under higher budgets as can be seen in Figure 3 left-(b). When we look at the proportion of wins by severity (Figure 3 right side), we notice a few critical improvements, most of which are attributed to sequential.

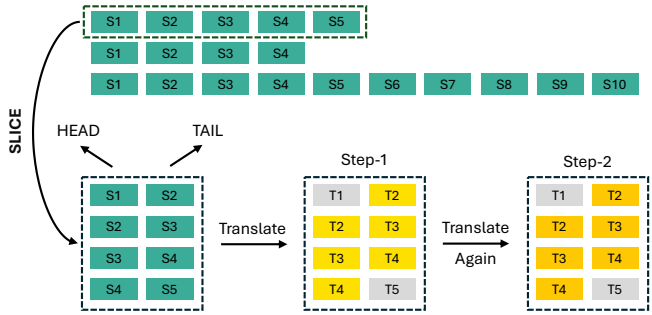
In Figure 3 left-(c), when we compare the sequential vs parallel, both under higher budgets (30), we observe that sequential substantially improves fluency and style metrics on average, but at the cost of reduced accuracy, compared to parallel. A possible explanation for the lower accuracy of the sequential approach is that, as multi-turn generation progresses, the influence of the source sentence gradually weakens, leading to decreasing faithfulness to the source. Our manual review of the samples also suggests that sequential rephrasing can be quite creative, as models embrace the variability of natural language, which might pose challenges for selector models.

6.2 On the Success of "Translate Again"

As shown in Figure 2, sequential sampling is particularly effective in the early steps. A key question is what drives these improvements. Previous studies in LLM community attribute this to the "reflection" abilities (Madaan et al., 2023b; Guo et al., 2025; Muennighoff et al., 2025), although the underlying mechanism remains unclear and is often vaguely ascribed to human-like "intelligence". Here, we have a more concrete explanation.

Language model generation follows an autoregressive, left-to-right process, which naturally induces an asymmetric context structure: within a span of generation, later generation steps have seen the previous translated outputs to their left, but the earlier generation steps have no access to their right context. This phenomenon has long been studied in context-aware/document translation (Voita et al., 2018, 2019a,b; Lopes et al., 2020).

We argue that part of the success of "Translate Again" stems from the same effect: During first-pass generation, earlier and later segments differ



	HEAD	TAIL	DELTA
STEP-1	**82.3**	**83.0**	**-0.7**
STEP-2	**83.7**	**83.9**	**-0.2**
STEP-3	83.7	83.8	-0.1
STEP-4	83.6	83.8	-0.2

Figure 4: Illustration of the controlled experiments in §6.2. Each document in WMT24++ is sliced into a batch of two consecutive sentences and translated using the "Translate Again" strategy. We compare the average translation performance at the HEAD and TAIL positions and report the results in the table on the right. It is clear that Step-2 recovers performance for the HEAD position. Note that we discard T_1 and T_5 to ensure that each sentence appears exactly once as a HEAD segment and once as a TAIL segment, respectively, thereby enabling strict comparability.

in quality due to an asymmetric context. In contrast, the second-pass translation has access to the full global context from the first pass, allowing it to recover and improve translation quality. To demonstrate and isolate the effect of the asymmetric context, we conduct a controlled experiment, presented in Figure 4:

1. We split each document into consecutive sentences using a sliding window of size 2 to form a dataset such that each sentence appears *exactly once* as a HEAD segment and *exactly once* as a TAIL segment,[6] ensuring strict comparability across positions. In practice, we use WMT24++ and maintain the sentence-to-document mapping by its metadata.

2. We then apply the "Translate Again" strategy to the resulting corpus and compare the average translation performance between the HEAD and TAIL positions.

We present the average COMET score results for `en-zh,ru,nl` in Figure 4, which show that: (1) Translations in the HEAD position clearly underperform their counterparts in the TAIL position during the first round, highlighting the performance gap introduced by asymmetric context; and (2) Translation quality in the HEAD position is improved and recovered to a comparable level to that of the TAIL position in the second round. More experimental details can be found in Appendix D.

These results are closely aligned with the preceding hypothesis of asymmetric context, highlighting the inherent limitations of left-to-right autoregression. Also, they provide a concrete explanation for at least part of the improvements observed in self-refinement under sequential scaling.

[6] As shown in Figure 4, we remove T_1 in the head position and T_5 in the tail position to strictly ensure this constraint.

6.3 Sequential sampling is more diverse for smaller budgets, but saturates faster.

To better understand the plateau effect of sequential sampling for larger sample budgets, we conduct an output space diversity analysis, measuring the sample-effectiveness of the later rounds of generation. To compare the differences in output variability, we estimate a rate of marginally novel samples. When adding one sample y at a time, given a history of samples $H = \{y^1, \ldots, y^i\}$, we compare the new sample to each sample in the history pool (given a current budget value). We define an exact match distance as $D(y|H) = \min_{y^i \in H} [y \neq y^i]$, where $D(y|H) = 0 \iff$ the new output matches at least one of the existing samples in the history and $D(y|H) = 1$ means the new sample is "novel". For each budget value, we report the $D(y|H)$, averaged over the dataset. Our results in Figure 5 show that for earlier rounds, sequential sampling generates much fewer exact matches, starting from nearly zero duplicate rate for budget-2, whereas parallel already starts with a ~ 0.9 duplicate rate. For the later rounds, sequential sampling appears to produce more duplicates and show signs of the repetition bias: it starts to repeat the previous generations more often than parallel. In §7, we revisit this effect, showing that by controlling the context shaping, we can reduce the repetition bias.

7 Ablation study

Not all models benefit equally well from reflection. To complement the results with `Qwen-3-32B-Instruct`, we run the experi-

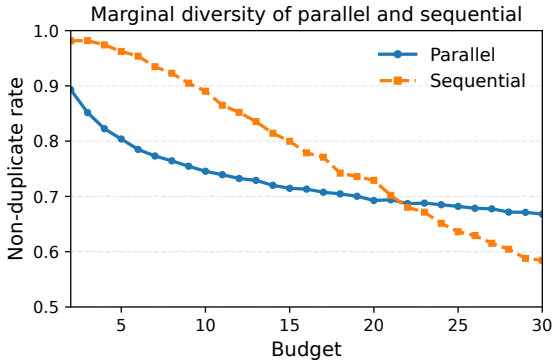


Figure 5: Averaged over 6 languages, diversity vs. budget for `Qwen-3-32B-Instruct` model. Non-duplicate rate for each budget is computed *w.r.t.* previously sampled outputs, converging to 0 for higher budget limits. Error bars, representing standard error of the mean ($\pm 1SE$), are nearly indistinguishable.

ments for a smaller model from the same family. Figure 10 presents the results for `Qwen-3-4B-Instruct`. We observe a similar overall pattern, *i.e.*, *sequential* achieves better performance under smaller budgets. However, its advantage over *parallel* is less pronounced than in larger models, which might indicate that small models have worse ability for multi-turn improvement. The "Translate Again" quality for this model can even go down (see en-zh results).

Larger sample history does not imply better performance. Our results in Figure 2 and analysis in §6.3 suggest that too long contexts of sequential sampling may lead to diminishing quality gains and repetition bias. To study the context effect further, we propose a sequential setting by capping the maximal number of samples that we store in context, such that for any sampling round, the number of samples stored in history is bound from above. In particular, we use a simple *sliding window* approach, setting h as the window size parameter:

$$y_h^{(t)} \sim p_\theta \left(\cdot | I, y^{(t-1)}, \ldots, y^{(t-h)}, I, y^{(1)}, I(x) \right). \quad (3)$$

To better align our experiments with parallel sampling, we always keep the initial translation $y^{(1)}$, and slide over the next h stored translations.

Table 2 results show that perhaps counterintuitively, larger history sizes do not imply better performance. The results for sequential with controlled history are better for almost every metric and language direction. Even more, smaller history sizes h lead to less COMET degradation for the self-refinement mode (Figure 15). From a sampling effectiveness angle, this result is particularly positive, as a smaller sequential history leads to both higher gains and lower costs. To explain this effect, we extend the diversity analysis (§6.3) to the history ablation experiment. From the Figure 7, we observe that controlling the history effectively reduces the repetition bias, which explains the better results of sequential with a controlled history. It might be promising to further study the context bias in the context of dynamical systems, looking for cycles or attractors (Wang et al., 2025b).

Sequential sampling does not collapse in the low temperature limit. For our main experiments, we compare parallel and sequential under a high-diversity/creativity regime using temperature 1.0. Temperature sampling, while increasing diversity, is well known to incur more errors due to random sampling (Troshin et al., 2025a). We expect a good selector model to effectively ignore samples with translation errors. Empirically, we observe better results with lower temperature for sequential sampling without a selector (Figure 18), which highlights this phenomenon.

When decreasing the temperature of the parallel sampling, eventually all diversity is expected to vanish, when approaching the greedy decoding. Sequential sampling, however, has the potential to remain useful for Best-of-N sampling even under extremely low temperature mode (including greedy decoding). Figure 17 confirms this hypothesis, showing the results for sequential with selector comparing greedy sampling vs high-temperature sampling. This observation may be useful for future work on developing better sampling methods, that optimize the quality-diversity trade-offs.

8 Conclusion

We conducted a set of experiments on open-source and closed-source models to compare two widely used paradigms for test-time scaling: sequential and parallel. Our results highlight the superiority of the sequential paradigm in terms of sample efficiency and low output-space duplicate rate. Our analysis and ablation study partially explain the success of sequential sampling through the lens of context processing; we further underline the importance of context control, demonstrating more stable sequential sampling under context-constrained scenarios, paving a road for further improvement.

9 Limitations

One limitation of our work is that we test against a single prompt format, even though our prompting strategy is motivated by previous work (Wu et al., 2025a). We expect that our findings can change, if one replaces the "Please translate again for a better version" prompt with other types of refinement instructions, for example purely diversity focused. Moreover, we observe that the sequential reflection abilities of different models can vary, thus our findings may or may not generalize to other models, depending on the model's capabilities.

Another limitation is that we tested our approach using a single dataset and only a small subset of languages. Using only a single dataset is somewhat balanced out given that WMT24++ covers a broad range of domains and language styles. We didn't test on low-resource or extremely low-resource languages, prioritizing covering a set of languages, for which we have access to human annotators. Our MQM analysis only targets 3 directions of translation quality, and does not cover other directions such as terminology use or locale conventions. For every experiment, due to resource constraints, we only sampled a single trajectory for each sequential run, which limits the stability of the reported quality values.

Finally, our findings regarding the "Why" questions are limited to the particular methodological strategies we used to monitor the performance differences. Our diversity analysis is limited to hard matches (duplicate rate), and can be extended using soft-matching utility functions for diversity. Our history control analysis only analyses a single strategy to compact the context, namely sliding window approach.

10 Ethical considerations

We acknowledge that all generation settings discussed in this work can be biased towards or against certain attributes, can systematically delete or suppress certain aspects of translations. In particular, our results under Best-of-N setup inherit the biases of the selector metric, and are shaped by what kind of translations are picked up by the selector metric. Even if the selector metric would be perfectly unbiased, in the setting of limited history size, it is possible that the LLM has biases, steering its improvements away from certain aspects.

References

Sourya Basu, Govardana Sachitanandam Ramachandran, Nitish Shirish Keskar, and Lav R. Varshney. 2021. Mirostat: A neural text decoding algorithm that directly controls perplexity. In *ICLR*.

Eleftheria Briakou, Jiaming Luo, Colin Cherry, and Markus Freitag. 2024. Translating step-by-step: Decomposing the translation process for improved translation quality of long-form texts. In *Proceedings of the Ninth Conference on Machine Translation*, pages 1301–1317, Miami, Florida, USA. Association for Computational Linguistics.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan O Arik. 2025. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*.

Gonçalo M. Correia and André F. T. Martins. 2019. A simple and effective approach to automatic post-editing with transfer learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3050–3056, Florence, Italy. Association for Computational Linguistics.

Haikang Deng and Colin Raffel. 2023. Reward-augmented decoding: Efficient controlled text generation with a unidirectional reward model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11781–11791, Singapore. Association for Computational Linguistics.

Daniel Deutsch, Eleftheria Briakou, Isaac Rayburn Caswell, Mara Finkelstein, Rebecca Galor, Juraj Juraska, Geza Kovacs, Alison Lui, Ricardo Rei, Jason Riesa, Shruti Rijhwani, Parker Riley, Elizabeth Salesky, Firas Trabelsi, Stephanie Winkler, Biao Zhang, and Markus Freitag. 2025. WMT24++: Expanding the language coverage of WMT24 to 55 languages & dialects. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12257–12284, Vienna, Austria. Association for Computational Linguistics.

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.

Zhaopeng Feng, Yan Zhang, Hao Li, Bei Wu, Jiayu Liao, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2025. TEaR: Improving LLM-based machine translation with systematic self-refinement. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3922–3938, Albuquerque, New Mexico. 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, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics*, 9:1460–1474.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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 *ICLR*.

Yichong Huang, Baohang Li, Xiaocheng Feng, Wenshuai Huo, Chengpeng Fu, Ting Liu, and Bing Qin. 2024. Aligning translation-specific understanding to general understanding in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 5028–5041, Miami, Florida, USA. Association for Computational Linguistics.

Evgenia Ilia and Wilker Aziz. 2024. Predict the next word: <humans exhibit uncertainty in this task and language models _____>. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 234–255, St. Julian's, Malta. Association for Computational Linguistics.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Dayeon Ki and Marine Carpuat. 2024. Guiding large language models to post-edit machine translation with error annotations. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4253–4273, Mexico City, Mexico. Association for Computational Linguistics.

Klaus-Rudolf Kladny, Bernhard Schölkopf, and Michael Muehlebach. 2025. Conformal generative modeling with improved sample efficiency through sequential greedy filtering. In *The Thirteenth International Conference on Learning Representations*.

Kevin Knight and Ishwar Chander. 1994. Automated postediting of documents. In *Proceedings of the Twelfth AAAI National Conference on Artificial Intelligence*, AAAI'94, page 779–784. AAAI Press.

Kasia Kobalczyk, Nicolás Astorga, Tennison Liu, and Mihaela van der Schaar. 2025. Active task disambiguation with LLMs. In *The Thirteenth International Conference on Learning Representations*.

Tom Kocmi, Ekaterina Artemova, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Konstantin Dranch, Anton Dvorkovich, Sergey Dukanov, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Howard Lakougna, Jessica Lundin, Christof Monz, Kenton Murray, and 10 others. 2025. Findings of the WMT25 general machine translation shared task: Time to stop evaluating on easy test sets. In *Proceedings of the Tenth Conference on Machine Translation*, pages 355–413, Suzhou, China. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, and 3 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, Miami, Florida, USA. Association for Computational Linguistics.

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.

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 E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

António Lopes, M. Amin Farajian, Rachel Bawden, Michael Zhang, and André F. T. Martins. 2020.

Document-level neural MT: A systematic comparison. In *Proceedings of the 22nd Annual Conference of the European Association for Machine Translation*, pages 225–234, Lisboa, Portugal. European Association for Machine Translation.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023a. Self-refine: Iterative refinement with self-feedback. *Preprint*, arXiv:2303.17651.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, and 1 others. 2023b. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594.

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 *ICLR*.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, Jilin Chen, Alex Beutel, and Ahmad Beirami. 2024. Controlled decoding from language models. In *ICML*.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candes, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20275–20321, Suzhou, China. Association for Computational Linguistics.

Dojun Park and Sebastian Padó. 2024. Multidimensional machine translation evaluation: Model evaluation and resource for Korean. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 11723–11744, Torino, Italia. ELRA and ICCL.

Sara Rajaee, Sebastian Vincent, Alexandre Berard, Marzieh Fadaee, Kelly Marchisio, and Tom Kocmi. 2026. Unlocking reasoning capability on machine translation in large language models. *Preprint*, arXiv:2602.14763.

Ricardo Rei, Nuno M. Guerreiro, José Pombal, Daan van Stigt, Marcos Treviso, Luisa Coheur, José G. C. de Souza, and André F. T. Martins. 2023. Scaling up CometKiwi: Unbabel-IST 2023 submission for the quality estimation shared task. In *Proceedings of the Eighth Conference on Machine Translation*, pages 841–848, Singapore. Association for Computational Linguistics.

Irina Saparina and Mirella Lapata. 2024. Ambrosia: A benchmark for parsing ambiguous questions into database queries. In *Advances in Neural Information Processing Systems*, volume 37, pages 90600–90628. Curran Associates, Inc.

Irina Saparina and Mirella Lapata. 2025. Disambiguate first, parse later: Generating interpretations for ambiguity resolution in semantic parsing. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 16825–16839, Vienna, Austria. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for 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 177–184, Boston, Massachusetts, USA. Association for Computational Linguistics.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 203–206, Prague, Czech Republic. Association for Computational Linguistics.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. In *Advances in Neural Information Processing Systems*, volume 35, pages 9460–9471. Curran Associates, Inc.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*.

Zayne Sprague, Fangcong Yin, Juan Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations*, volume 2025, pages 94118–94162.

Shaomu Tan, Ryosuke Mitani, Ritvik Choudhary, and Toshiyuki Sekiya. 2025. Investigating test-time scaling with reranking for machine translation. *arXiv preprint arXiv:2509.19020*.

Shaomu Tan and Christof Monz. 2025. ReMedy: Learning machine translation evaluation from human preferences with reward modeling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4370–4387, Suzhou, China. Association for Computational Linguistics.

Shaomu Tan, Dawei Zhu, Ke Tran, Michael Denkowski, Sony Trenous, Bill Byrne, Leonardo Ribeiro, and Felix Hieber. 2026. What does llm refinement actually improve? a systematic study on document-level literary translation. In *ACL*.

Sergey Troshin, Wafaa Mohammed, Yan Meng, Christof Monz, Antske Fokkens, and Vlad Niculae. 2025a. Control the temperature: Selective sampling for diverse and high-quality LLM outputs. In *Second Conference on Language Modeling*.

Sergey Troshin, Irina Saparina, Antske Fokkens, and Vlad Niculae. 2025b. Asking a language model for diverse responses. In *Proceedings of the 2nd Workshop on Uncertainty-Aware NLP (UncertaiNLP 2025)*, pages 66–72, Suzhou, China. Association for Computational Linguistics.

Luke Vilnis, Yury Zemlyanskiy, Patrick Murray, Alexandre Passos, and Sumit Sanghai. 2023. Arithmetic sampling: parallel diverse decoding for large language models. In *ICML*.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019a. Context-aware monolingual repair for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 877–886, Hong Kong, China. Association for Computational Linguistics.

Elena Voita, Rico Sennrich, and Ivan Titov. 2019b. When a good translation is wrong in context: Context-aware machine translation improves on deixis, ellipsis, and lexical cohesion. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1198–1212, Florence, Italy. Association for Computational Linguistics.

Elena Voita, Pavel Serdyukov, Rico Sennrich, and Ivan Titov. 2018. Context-aware neural machine translation learns anaphora resolution. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1264–1274, Melbourne, Australia. Association for Computational Linguistics.

George Wang, Jiaqian Hu, and Safinah Ali. 2025a. Maats: A multi-agent automated translation system based on mqm evaluation. *Preprint*, arXiv:2505.14848.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024. MMLU-pro: A more robust and challenging multi-task language understanding benchmark. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhilin Wang, Yafu Li, Jianhao Yan, Yu Cheng, and Yue Zhang. 2025b. Unveiling attractor cycles in large language models: A dynamical systems view of successive paraphrasing. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12740–12755, Vienna, Austria. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Di Wu, Seth Aycock, and Christof Monz. 2025a. Please translate again: Two simple experiments on whether human-like reasoning helps translation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20435–20451.

Di Wu, Yibin Lei, and Christof Monz. 2025b. Calibrating translation decoding with quality estimation on llms. In *Advances in Neural Information Processing Systems*, volume 38, pages 155660–155689. Curran Associates, Inc.

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. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. 2025. Verbalized sampling: How to mitigate mode collapse and unlock llm diversity. *Preprint*, arXiv:2510.01171.

A Prompts

A.1 Prompt for parallel

We provide the prompt used for parallel sampling here. Note that we explicitly instruct the model not to generate any additional content beyond the translation itself. In addition, we disable thinking mode by explicitly specifying this requirement in the system prompt. This prompt is shared with the first round of sequential sampling, ensuring strict comparability; see Appendix A.2.

```
[system]
You are a professional translator. You
will be asked to conduct
translation-related tasks. Output ONLY
the transaltion segment - do NOT generate
additional content. Do not include any
<think> tags. Do not show your reasoning.

[user turn initial]
Provide only one translation and do not
output anything else after that.
Translate the following text from English
to {target language}.

English: {source}

[assistant turn]
{assistant response}
```

A.2 Prompt for multi-turn sequential sampling (improving instruction)

We provide the multi-turn prompt used for sequential sampling here. Note that the prompt for the first round is identical to that in Appendix A.1, as shown in green. The iterative refinement instruction is shown in red, following Wu et al. (2025a), where we ask the model to "Translate Again" for a better version.

```
[system]
You are a professional translator. You
will be asked to conduct
translation-related tasks. Output ONLY
the transaltion segment - do NOT generate
additional content. Do not include any
<think> tags. Do not show your reasoning.

[user turn initial]
Provide only one translation and do not
output anything else after that.
Translate the following text from English
to {target language}.

English: {source}

[assistant turn]
{assistant response}
```

```
[user turn revisiting]
Please translate again for a better
version.

[assistant turn]
{assistant response}

[user turn revisiting]
Please translate again for a better
version.

...
```

B Generation Details

For the open-source models, we generate samples using `vLLM v0.11.1` (Kwon et al., 2023) inference engine. We run `Qwen3-32B-Instruct` on 4 `NVIDIA RTX A6000` cards with up to 8 request processing at the same time. With the total budget 30, each full-history sequential run, for a single language pair takes on average 14h; each history-1 sequential sampling run takes slightly more that 4h, and each parallel run takes on average 4h. We run `Qwen3-4B-Instruct-2507` on 2 `NVIDIA RTX A6000` cards processing on average 40 requests in parallel. History-5 sequential run takes on average 1h 18m, and each parallel run takes on average 20m. Both models are run with the generation parameters, reported in Table 1.

We run the `GPT-4o-mini` using official API from the model provider, using default generation parameters for this model (no top-p, no top-k filtering, temperature 1.0). For this model, we slightly edit the prompt, by replacing "English: {source}" line simply with "{source}", otherwise model some-

Parameter	Value
max-model-len	32768
max-num-batched-tokens	16392
max-num-seqs	8
max-tokens	4096
top-k	20
top-p	0.8

Table 1: `Qwen3-*` generation parameters.

times includes target language name in its outputs.

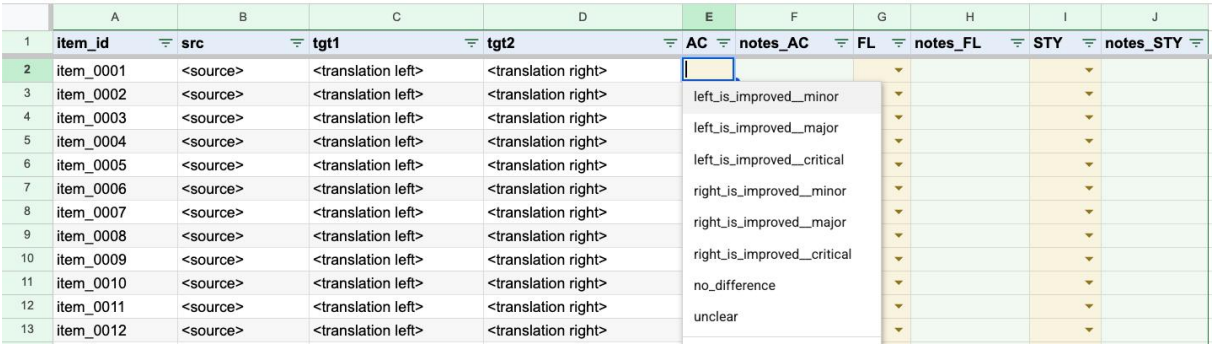
We run the evaluation models on a single `NVIDIA RTX A6000` GPU. We use batch size 8 for COMET (`wmt22-comet-da`), COMET-KIWI (`wmt23-cometkiwi-da-xl`), and MetricX (`metricx-24-hybrid-xxl-v2p6-bfloat16`) metrics, and batch size 32 for the ReMedy (`ReMedy-9B-22`) metrics. For MetricX, we set max input length as 1536, and for ReMedy as 4096. For ReMedy, we do not use callibration, as this feature is needed only to spread out the score distribution. Other metric parameters are as default.

C Human Annotation

Annotators are native speakers of a target language (researchers, students, university workers), and participate on a voluntary, unpaid basis. They were provided with the labeling instructions(Appendix C.1, and then asked to annotate which one of the two translations is better for each quality dimension: accuracy, fluency, and style, given the source sentence in English. For our pilot human annotation experiments, we use a blind comparison setup, *i.e.*, annotators see a shuffled pairs of target segments. We randomly select 100 instances from the WMT24++ segment-level dataset for each annotation experiment. Source sentence ids are the same across target languages for each experiment type, but the source sentences do not intersect between different experiments to prevent memorization biases and model guessing. We use a simple tabular interface via Google Sheets as a tool for collecting labels, where annotators select the labels using a predefined set of labels defined in Appendix C.1, and can optionally leave the notes, explaining their decisions. We include a screenshot of our interface in Figure 6.

C.1 Human annotation instructions

```
Use the source text as the reference and
compare translation1 and translation2 for
each MQM dimension separately: AC accuracy,
FL fluency, and STY style. Assign one label
per dimension.

Choose exactly one label:

left_is_improved_minor
left_is_improved_major
left_is_improved_critical
right_is_improved_minor
right_is_improved_major
right_is_improved_critical
no_difference
unclear

Here, left = translation1 and right =
translation2.

Use left_is_improved if translation1 is
better.
Use right_is_improved if translation2 is
better.
Use no_difference if both translations are
equivalent for that dimension, including
when both are equally good or equally bad.
Use unclear only when the source or
translations do not provide enough
information to make a reliable judgment, or
both translations have positive and negative
aspects.

Severity:
minor = small improvement that does not
substantially affect meaning, fluency, or
appropriateness.
major = clear improvement that noticeably
affects meaning, fluency, or style.
critical = improvement fixes or avoids a
severe issue that makes the translation
seriously misleading, unusable, or
inappropriate.

AC — Accuracy: judge whether the translation
correctly preserves the source meaning.
Consider distortions, omissions, and
additions.
FL — Fluency: judge linguistic
well-formedness in the target language.
Consider grammar, syntax, spelling,
punctuation, and mechanical correctness.
STY — Style: judge whether the language
style/use is appropriate for the context,
even if grammatically correct.
```



	A	B	C	D	E	F	G	H	I	J
1	item_id	src	tgt1	tgt2	AC	notes_AC	FL	notes_FL	STY	notes_STY
2	item_0001	<source>	<translation left>	<translation right>						
3	item_0002	<source>	<translation left>	<translation right>						
4	item_0003	<source>	<translation left>	<translation right>						
5	item_0004	<source>	<translation left>	<translation right>						
6	item_0005	<source>	<translation left>	<translation right>						
7	item_0006	<source>	<translation left>	<translation right>						
8	item_0007	<source>	<translation left>	<translation right>						
9	item_0008	<source>	<translation left>	<translation right>						
10	item_0009	<source>	<translation left>	<translation right>						
11	item_0010	<source>	<translation left>	<translation right>						
12	item_0011	<source>	<translation left>	<translation right>						
13	item_0012	<source>	<translation left>	<translation right>						

Figure 6: A demonstration of the human annotation interface using Google Sheets. For visibility of the demonstration, we replace the exact sentences with the placehoders for the source, left and right translations. Translations are presented in a shuffled way to mitigate the position bias for human annotators.

D Experimental Details for §6.2

We conduct our experiments on `Qwen-3-32B-Instruct` model. The "Translate Again" setting is identical to that in §5.

The WMT24++ dataset consists of 171 documents and 998 sentences (some of which are short paragraphs). We use the metadata[7] provided by WMT24++ to preserve the sentence-to-document mapping. Based on this mapping, we construct 633 segments (two consecutive sentences) following a sliding window setting, as described in §5. For each document, we remove one HEAD and one TAIL sentence to ensure strict comparability.

Note that we use a special separator "\t" to connect HEAD and TAIL sentences in each segment and split them when calculating the corresponding sentence-level score.

E Diversity Results

In Figure 8, we show the non-duplicate rate (marginal diversity) per language direction. Results demonstrate the uniformity of the diversity findings over the language directions: sequential produce less duplicates compared to parallel for earlier rounds of generation.

In Figure 7, we show the non-duplicate rate per history ablation experiment, demonstrating that compacting the history can help to avoid the bias towards higher duplicate rate.

F Results

Qwen3-32b-instruct results. Figure 2 presents main results across 6 language pairs for the `Qwen3-32b-instruct` model.

[7] https://huggingface.co/datasets/google/wmt24pp

Table 2 presents a compact view to easier compare different strategies 1) first steps of "Translate Again", 2) sampling with sequential selector results, including the history window size ablation (h = 'full', $h = 5$, $h = 1$).

Qwen3-4b-instruct additional results. Results for the `Qwen3-4b-instruct` are presented in Figure 10.

GPT-4o-mini additional results. Results for the `GPT-4o-mini` are presented in Figure 11.

G Human Evaluation Results

For the human evaluation, we present the full results with a split by language directions in Figure 3. When comparing sequential w selector, or sequential turn-2 vs turn-1, we notice, that sequential sampling improves fluency, accuracy and style; but the quality improvements can vary across languages and quality dimensions. When comparing sequential vs parallel under budget 30, we notice that parallel often demonstrates higher accuracy. We hypothesize that this effect can be attributed to the relative position of the turn-1 vs other turns, *i.e.*, sequential outputs are positioned closer to the source sentence, which preserves source information better.

H Ablation Results

H.1 History ablation

We present the results of context managing for sequential sampling using the sliding window approach with history sizes: full history, history-5, history-2, history-1. Results without selector show substantial scaling improvements in Figure 15; Figure 16 presents results for sequential sampling with

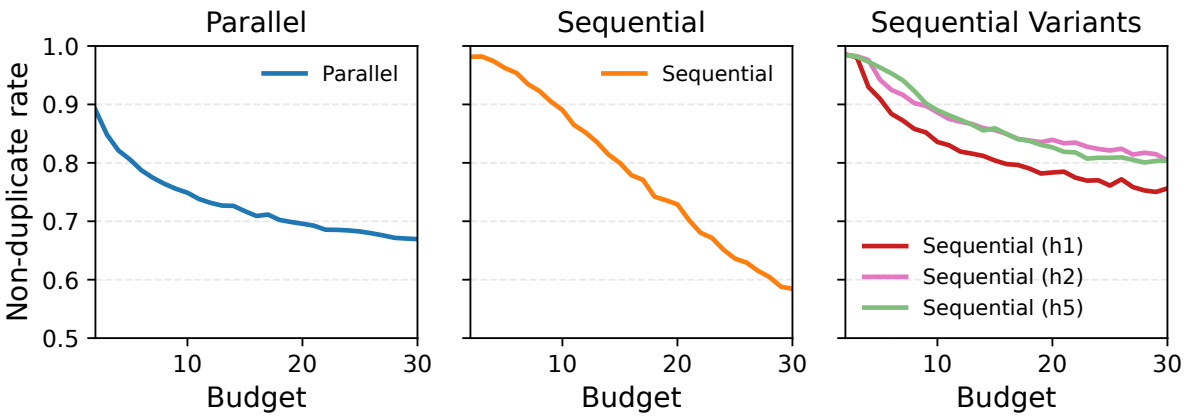


Figure 7: Non-duplicate rate for `Qwen-3-32B-Instruct` model for the history size ablation experiment for sequential (parallel is added for reference). Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

selector. Figure 7 shows the effect of output diversity for the history experiment, highlighting that capping the history helps to reduce the repetition bias.

H.2 Sampling hyperparameters

We use temperature 1.0 for all experiments, unless stated otherwise. Figure 18 visualizes quality vs budget scaling usingBest-of-N sampling, showing that temperature 1.0 is near optimal for Best-of-N sampling setting. We acknowledge that without Best-of-N sampling, lower temperature should be preferred to prioritize accuracy instead of diversity; however, we focus our experiments on the reranking setup due to its higher translation quality and scaling behavior.

Other sampling parameters are fixed for all experiments (top-k with $k = 20$, top-p with 0.8 as Qwen3 technical report recommends (Yang et al., 2025)). We set the maximal number of generated tokens per turn as 4096 for all experiments except full history experiments, where we set it to 764 to prevent model length overflows (given a small number of collapsed extra long outputs).

Lang	Metric	Seq turn-1	Seq turn-2	Seq turn-3	Par Selector	Seq Selector	Seq-h5 Selector	Seq-h1 Selector
en-de	COMET	81.26	81.59	81.49	**83.04**	82.62	82.24	82.84
	metricx24 (ref)	2.23	1.92	1.81	1.70	1.55	1.52	**1.46**
	metricx24 (qe)	2.02	1.72	1.62	1.53	1.38	1.36	**1.32**
en-nl	COMET	82.10	82.81	82.70	84.30	83.92	84.02	**84.52**
	metricx24 (ref)	3.94	3.33	3.16	2.87	2.57	2.47	**2.42**
	metricx24 (qe)	3.63	3.00	2.85	2.60	2.26	2.18	**2.12**
en-ru	COMET	80.98	82.44	82.35	83.21	83.70	83.51	**84.00**
	metricx24 (ref)	4.41	3.60	3.45	3.29	2.88	2.82	**2.73**
	metricx24 (qe)	3.86	3.09	2.96	2.76	2.41	2.31	**2.22**
en-zh	COMET	84.85	85.14	85.08	**86.06**	85.40	85.59	85.79
	metricx24 (ref)	2.47	2.20	2.21	2.13	2.02	1.98	**1.92**
	metricx24 (qe)	2.39	2.14	2.14	2.07	1.93	1.90	**1.85**
en-ar	COMET	76.11	77.30	77.54	78.84	79.37	79.40	**79.69**
	metricx24 (ref)	4.92	4.25	3.99	3.64	3.10	**3.01**	3.03
	metricx24 (qe)	4.37	3.72	3.44	3.08	2.56	**2.49**	2.52
en-ro	COMET	82.68	83.41	83.60	85.34	84.65	84.84	**85.46**
	metricx24 (ref)	5.90	5.10	4.88	4.44	3.99	3.87	**3.80**
	metricx24 (qe)	5.39	4.65	4.37	4.02	3.47	3.40	**3.30**

Table 2: Automatic evaluation results for sequential and parallel sampling strategies across 6 language pairs for the `Qwen3-32B-Instruct` model. Left side presents results for turn-$1, 2, 3$, of sequential sampling, right side shows results with Best-of-N sampling with budget 30 for both sequential (full context, $h = 5$, $h = 1$) and parallel with `COMET-KIWI` selector. The best results in each row are marked with **bold**.

		turn1 / turn2				turn1 / seq. w\ sel				par. w\ sel / seq. w\ sel			
Lang.	Metric	minr	majr	crit	nd	minr	majr	crit	nd	minr	majr	crit	nd
en-de	Accuracy	10/5	3/5	1/1	41	7/3	2/1	0/1	42	10/6	4/1	2/0	29
	Fluency	9/6	1/3	0/0	51	4/5	1/6	0/0	42	3/6	1/0	0/0	47
	Style	7/17	2/2	0/0	41	5/10	1/0	0/0	43	9/11	1/0	0/0	36
en-nl	Accuracy	20/19	11/15	1/1	31	10/20	3/12	1/7	27	20/17	8/6	5/1	19
	Fluency	11/25	4/11	0/0	46	9/22	1/9	0/1	53	6/12	0/0	0/0	78
	Style	5/3	1/4	0/0	86	32/11	10/1	0/0	43	29/21	4/2	1/0	40
en-ru	Accuracy	12/6	5/5	0/3	59	9/11	8/7	0/4	54	18/4	6/12	1/1	51
	Fluency	3/4	3/9	1/3	68	0/8	3/15	1/3	68	0/6	2/8	0/0	82
	Style	10/19	2/7	0/0	56	16/17	2/17	1/0	43	15/17	3/11	0/1	52
en-zh	Accuracy	8/4	6/2	3/4	36	16/14	9/14	1/3	33	21/10	8/7	2/2	41
	Fluency	1/2	1/1	0/0	57	2/10	1/5	0/2	72	1/5	1/2	0/0	81
	Style	5/9	2/6	0/0	41	2/23	6/14	1/6	42	7/16	4/10	0/1	52
en-ar	Accuracy	6/11	6/16	3/3	53	4/10	1/13	1/9	61	10/7	4/14	0/0	61
	Fluency	5/17	3/6	0/1	65	5/16	1/22	0/2	52	5/11	3/10	0/0	70
	Style	5/6	1/4	0/0	81	0/2	0/1	0/0	96	1/7	0/1	0/0	90

Table 3: Raw MQM annotation counts for the directional minor improvements (minr), major improvements (majr), critical improvements (crit), and no difference (nd); */* formatting corresponds to the experiment names in the first row of the figure; */* formatting for the relative scores means <n times left is improved>/<n times right is improved>. Annotation was partially finished for en-de due to time constraints.

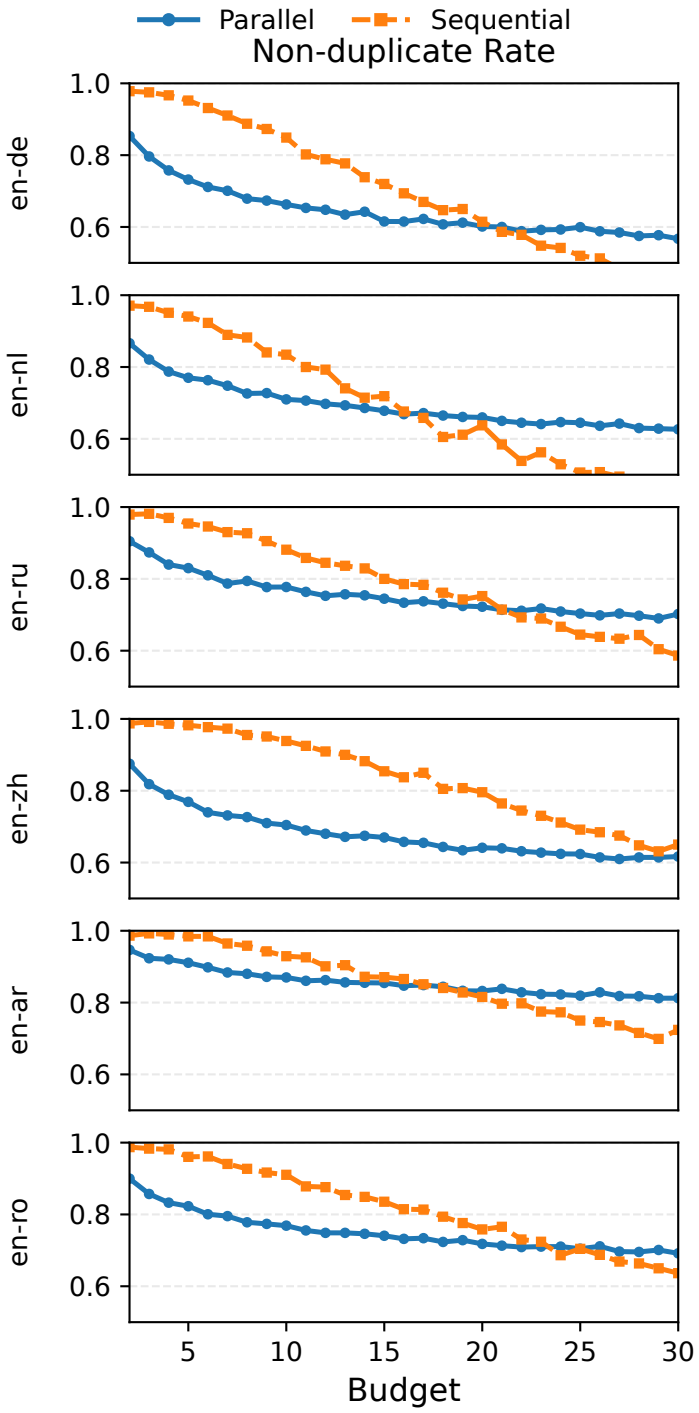


Figure 8: Non-duplicate rate for `Qwen-3-32B-Instruct` model for parallel and sequential sampling per language. Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

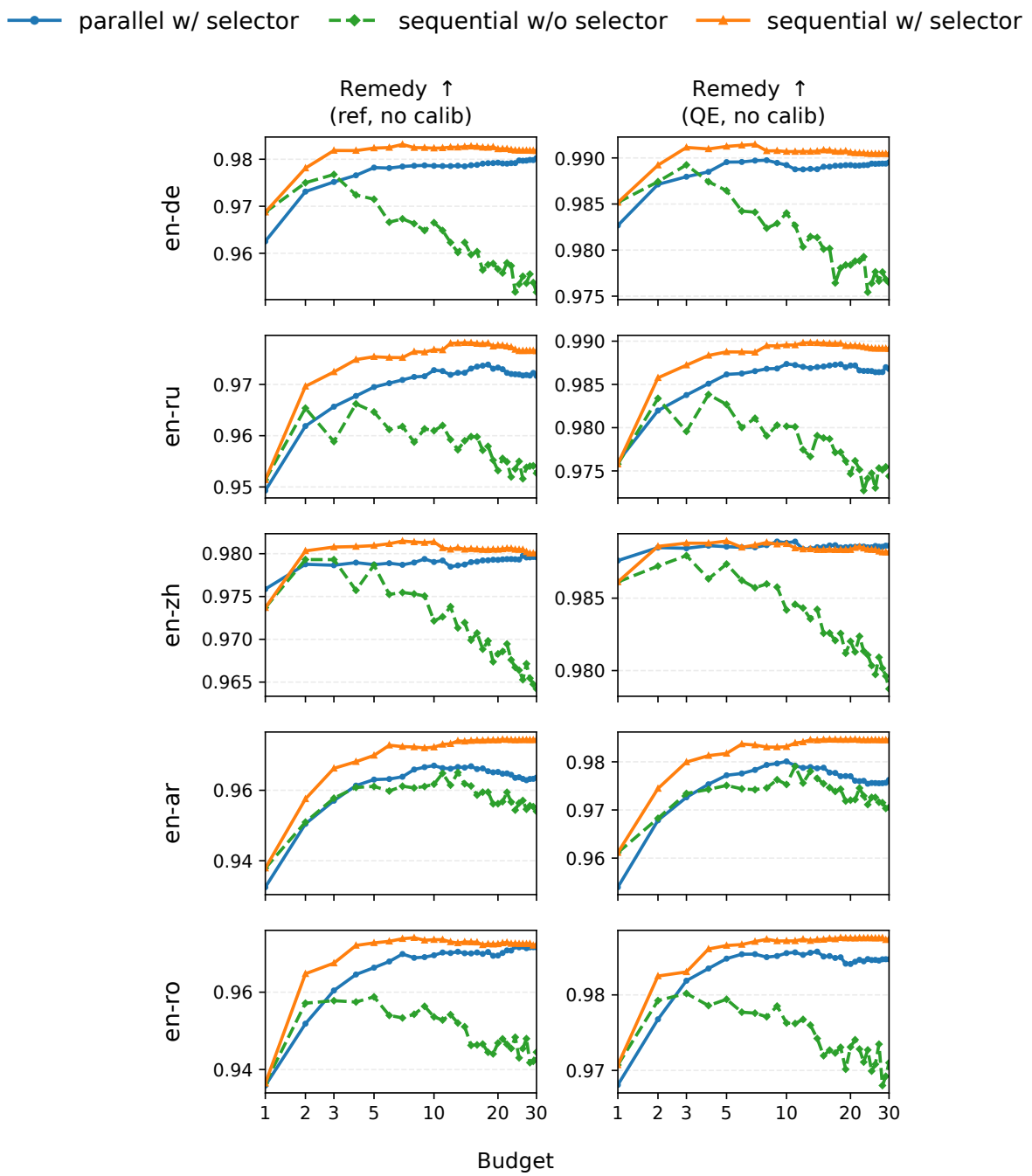


Figure 9: Additional evaluation with the ReMedy (`ReMedy-9B-22`) metric (Tan and Monz, 2025) for 5 language directions (en-nl is not supported by the metric). Samples are from the `Qwen-3-32B-Instruct` model. Quality vs. budget. Selector is `COMET-KIWI`.

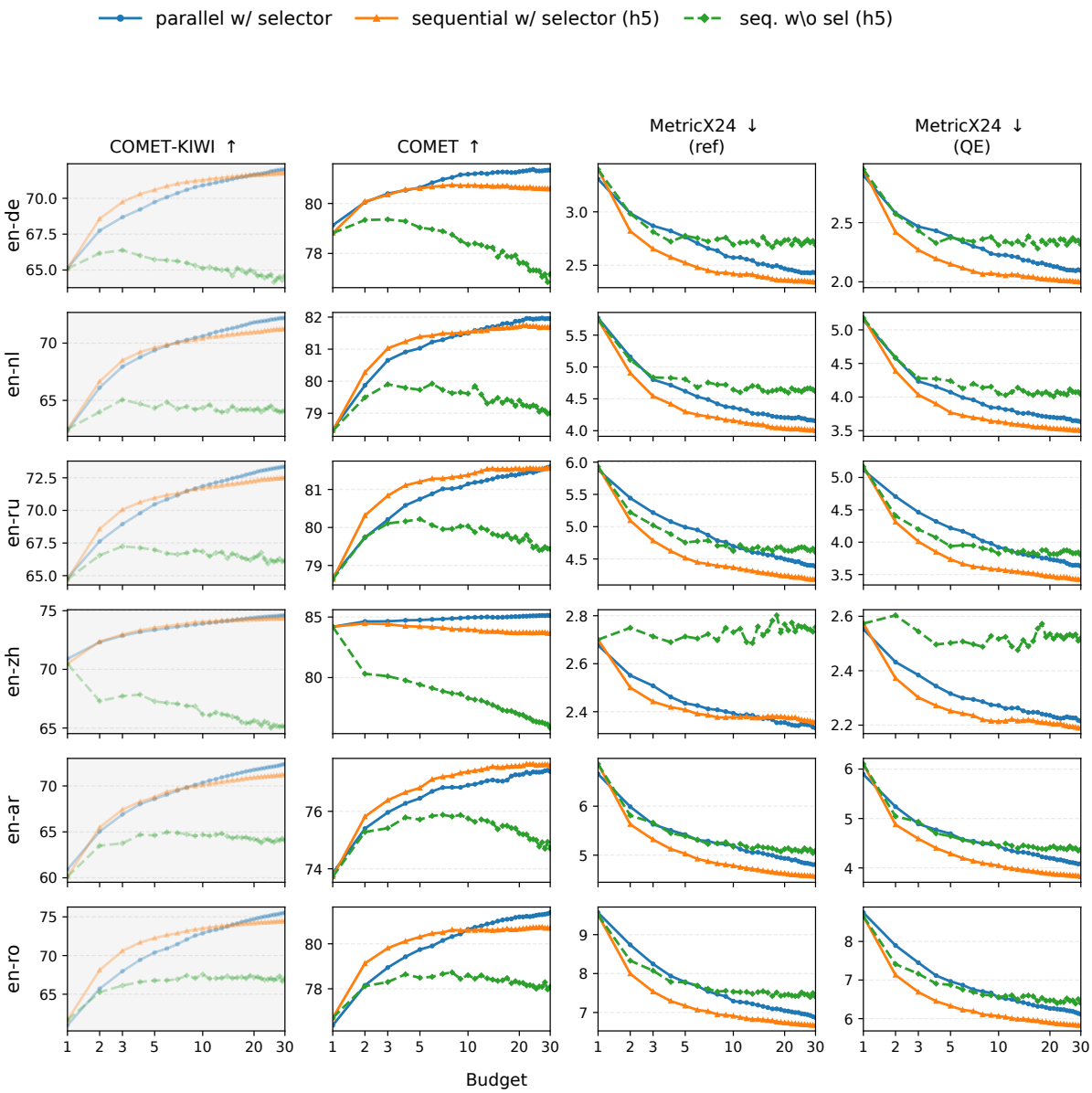


Figure 10: Quality vs budget results for the `Qwen-3-4B-Instruct-2507` model. Selector is `COMET-KIWI`.

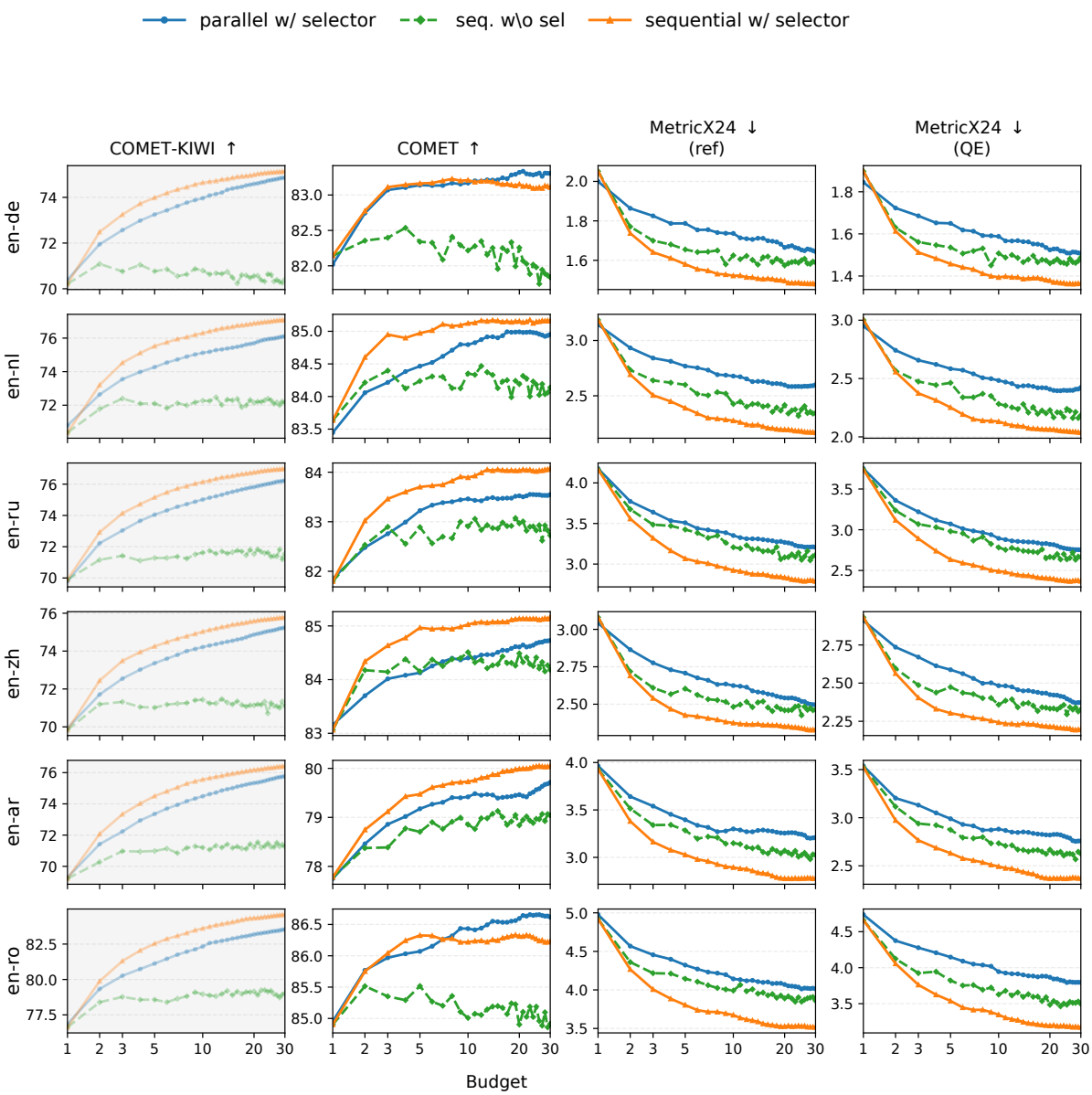


Figure 11: Quality vs budget results for the `GPT-4o-mini` model. Selector is `COMET-KIWI`.

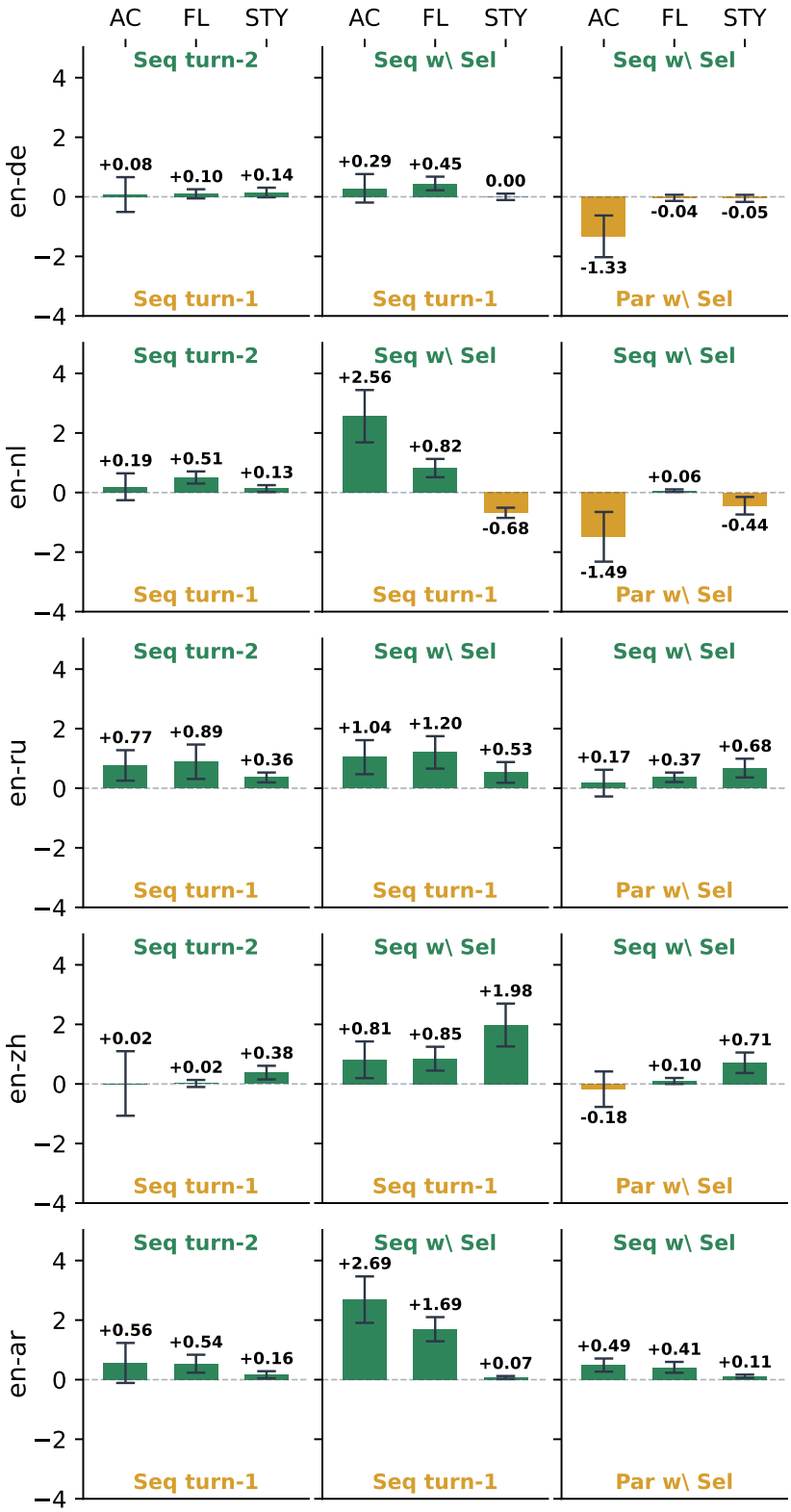


Figure 12: Human annotation results using MQM **average weighted relative improvement** for accuracy (AC), fluency (FL) and style (STY) for the for the `Qwen3-32B-Instruct` model. Results are presented for (i) turn-1 vs turn-2, (ii) Sequential w Sel (budget 30) vs Seq. turn-1, (iii) Sequential w Sel (budget 30) vs Parallel w Sel (budget 30). Error bars represent standard error of the mean.

Figure 13: Diversity and MT quality (COMET) for `Qwen-3-32B-Instruct` model for parallel sampling. Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

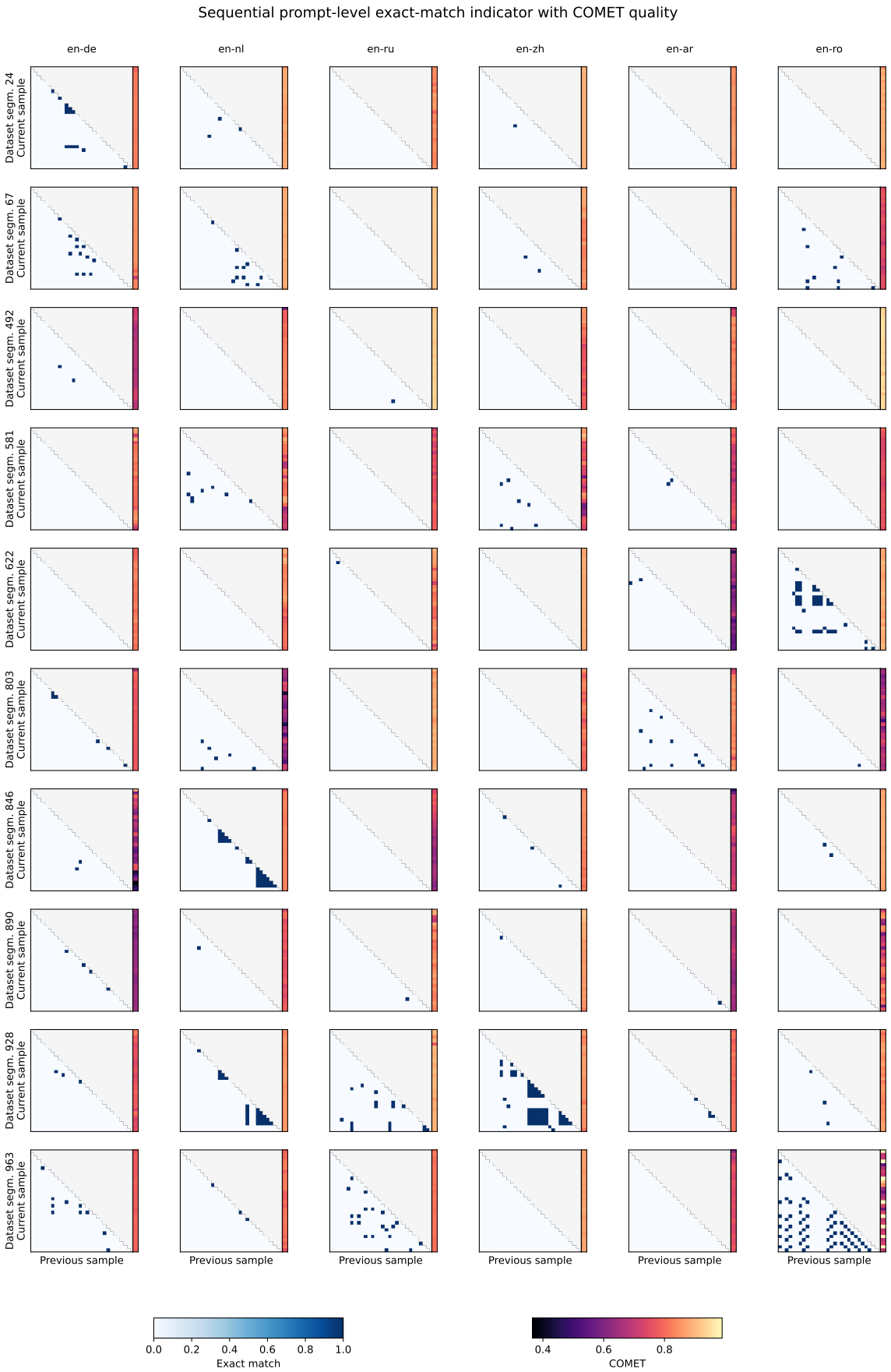


Figure 14: Diversity and MT quality (COMET) for `Qwen-3-32B-Instruct` model for sequential sampling. Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

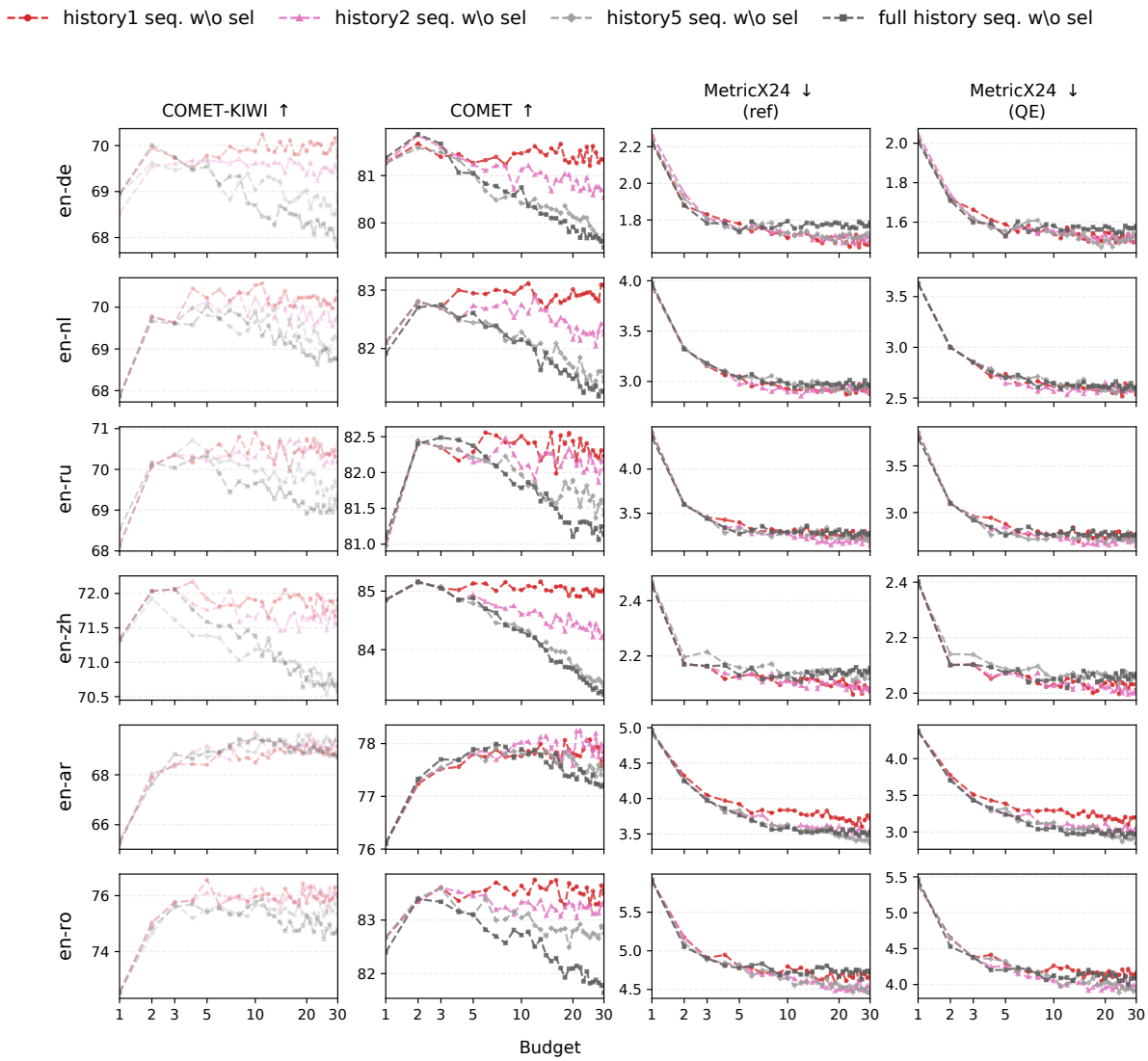


Figure 15: History size ablation for the sliding window sequential sampling approach for the `Qwen-3-32B-Instruct` model. The scaling of "Translate Again" can be largely improved by compacting the context using the sliding window approach.

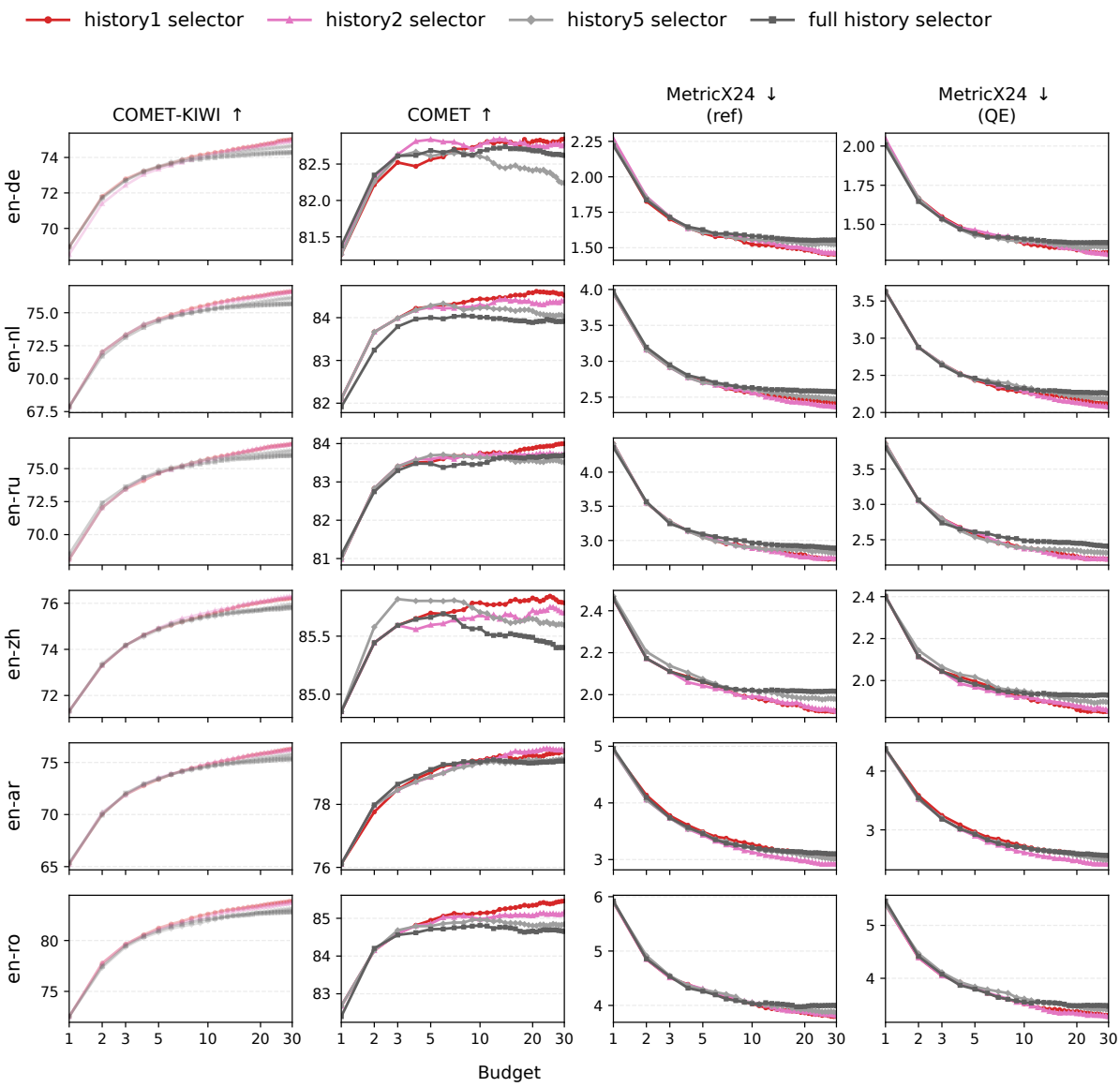


Figure 16: History size ablation for the sliding window sequential sampling approach for the `Qwen-3-32B-Instruct` model. The quality of sequential sampling with selector can be slightly improved by compacting the context using the sliding window approach.

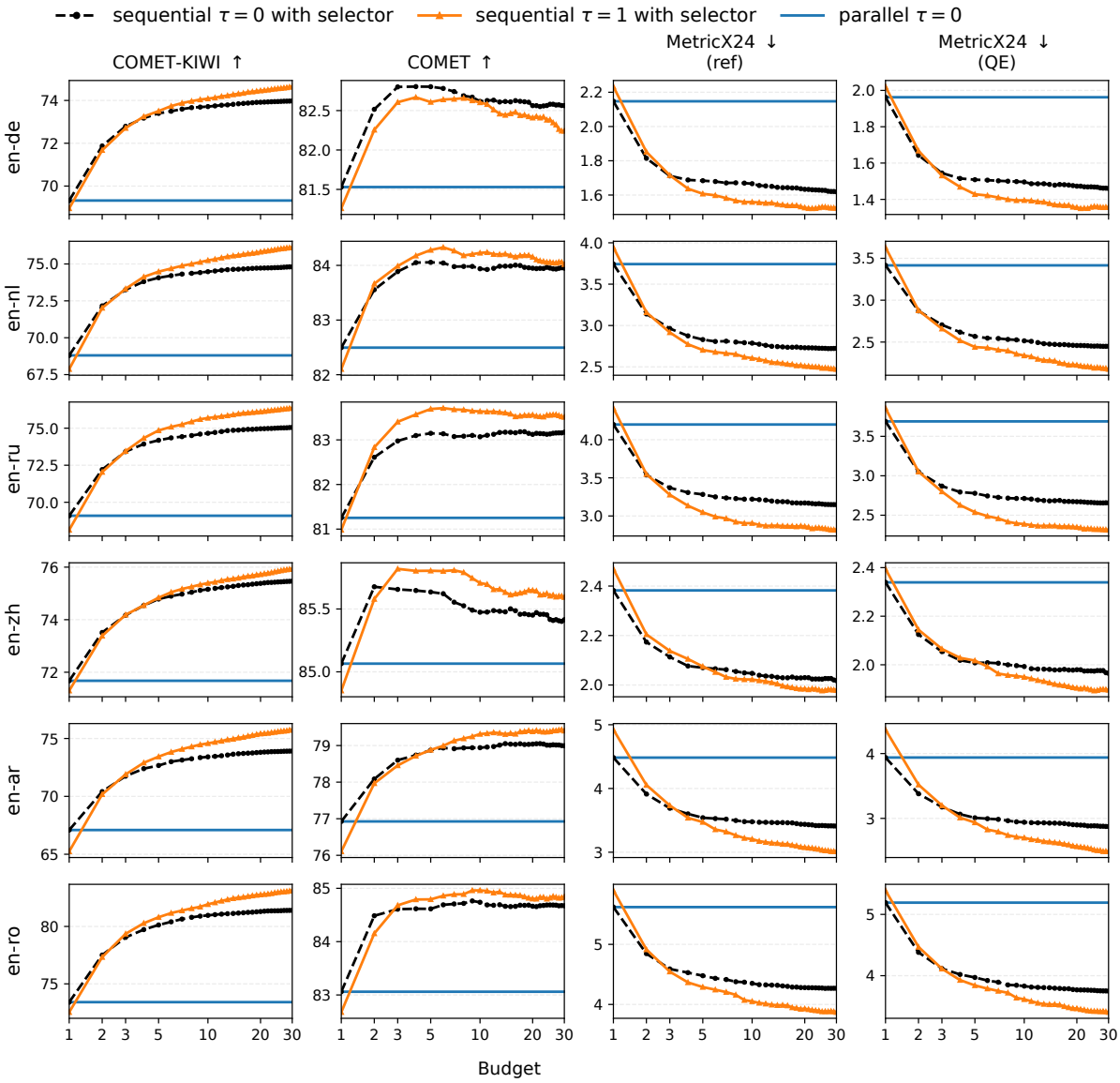


Figure 17: Temperature ablation for the sequential sampling approach. Sequential sampling demonstrates competitive results even for the greedy decoding (0.0 temperature regime), and is sometimes even better than the high-temperature variant for very small budgets. Parallel sampling in contrast is only applicable for higher temperatures.

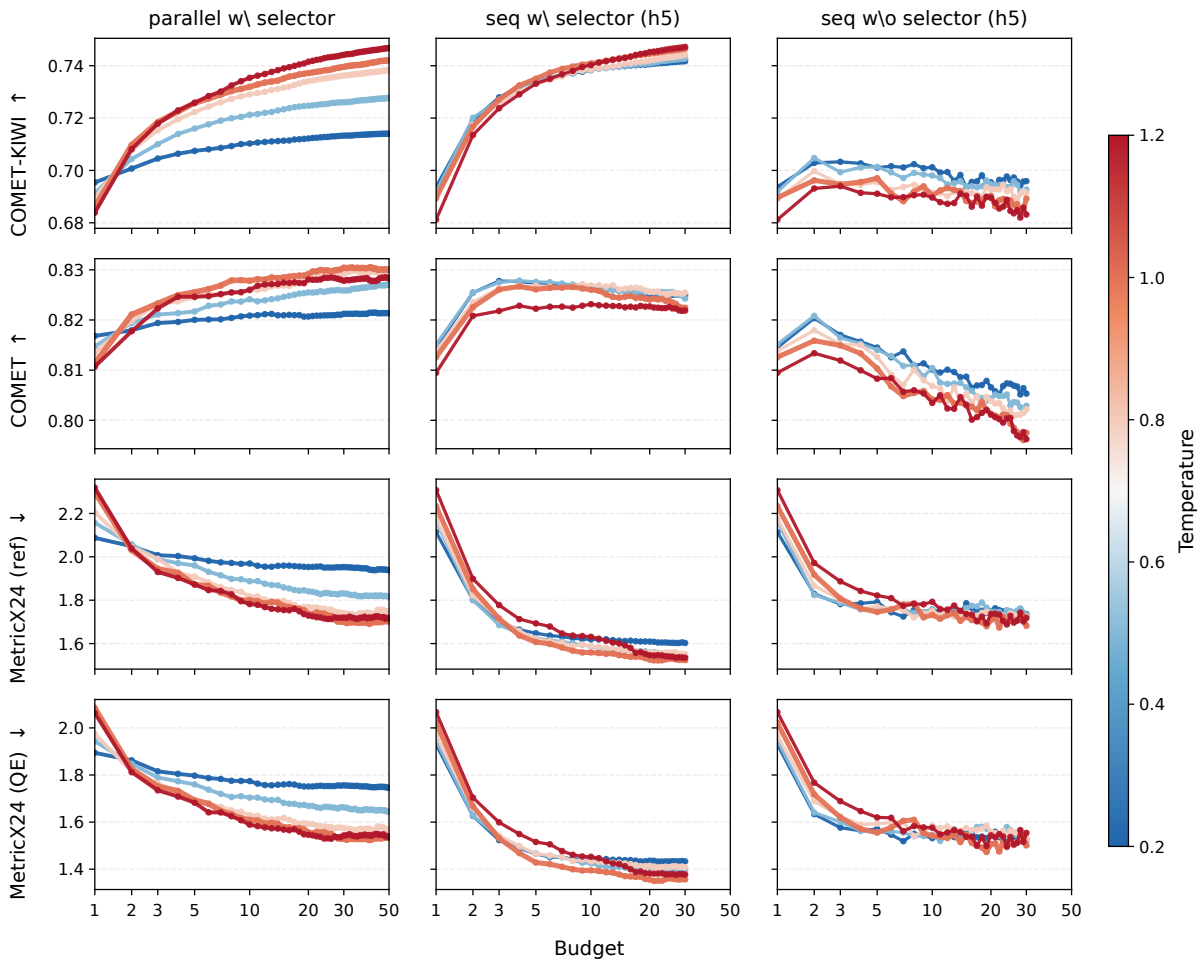


Figure 18: Temperature ablation for parallel and sequential runs for the en-de language pair. Temperature 1.0 is near optimal for the Best-of-N sampling results. For the "Translate Again" (no selector), lower temperature values lead to better results for the MT evaluation (high-precision, but low-diversity regime).