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

Di Wu[*,1] **Sergey Troshin**[*,1] **Christof Monz**[1] **Antske Fokkens**[2] **Vlad Niculae**[1]
[1]University of Amsterdam
[2]Vrije Universiteit Amsterdam
{d.wu,s.troshin,c.monz,v.niculae}@uva.nl
antske.fokkens@vu.nl

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 has a higher performance ceiling, providing a more diverse and effective pool of samples, particularly under smaller sampling 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 can degrade accuracy when inference budgets are large. 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 Models (LLMs) have been shown to benefit from scaling test-time computation (Snell et al., 2024; Muennighoff et al., 2025). In particular, iterative generation allows models to progressively refine their outputs based on previously generated responses, potentially yielding better solutions for many tasks, including mathematics and coding.

In Machine Translation (MT), LLM based systems represent the current state of the art in translation quality (Kocmi et al., 2024, 2025), where test-time scaling technology has also recently attracted increasing attention. Existing studies typically demonstrate its effectiveness through complex multi-agent systems or carefully designed reasoning pipelines (Briakou et al., 2024; Wang et al., 2025a). Although promising, their complexity makes it difficult to isolate the effects of individual components, thereby limiting generalizability and hindering a deeper understanding of the underlying mechanisms. For example, Wu et al. (2025a) show that simply prompting an LLM to translate again and refine can achieve even better performance than a complex, human-like reasoning pipeline.

To better understand the mechanisms underlying test-time scaling in machine translation, we study two commonly used paradigms in a maximally simplified and controlled setting: (1) **parallel**, for which the translations are *sampled* independently, followed by reranking using external metric model, *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*, as exemplified by (Wu et al., 2025a). Beyond machine translation, this paradigm has been interpreted as an advanced LLM capability, namely self-reflection (Madaan et al., 2023; Muennighoff et al., 2025).

These two representative sampling strategies explore the decoding space in orthogonal ways, at the cost of increased computational overhead. In this study, we control the sampling budget, i.e., the number of generation rounds, to be comparable across strategies and investigate: (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 mechanisms drive the performance differences; and (4) what is the effect of context shaping on sequential sampling. The outcome of our analyzes can be

[*]Equal first-author contribution.

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 51863–51893
October 24-29, 2026 ©2026 Association for Computational Linguistics

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 under larger budget.
- 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., 2023; Kamoi et al., 2024; Muennighoff et al., 2025).
- We further show that simply restricting the context to a single generation round in sequential sampling leads to clear performance gains across metrics, highlighting the double-edged role of context in sequential scaling: more context can be counterproductive.

2 Background

Test-time scaling. This strategy 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., 2023). 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 or previous generation 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., 2023).

In our study, we follow Wu et al. (2025a) and consider a minimal setting for self-refinement to emphasize simplicity (§3.2). Note that we employ an external metric model for Best-of-N selection solely to demonstrate the potential of the generated sample pool at evaluation time; *i.e.*, the metric signal is never used to guide the generation process.

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); and for 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) and Zhang et al. (2026) formulate variants of sequential sampling to generate diverse outputs from LLMs.

3 Sequential and Parallel Sampling

In this section, we formally define parallel and sequential sampling. Both sampling algorithms can be applied up to a fixed number of generation rounds (budget), ensuring fair comparison.

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 times 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 generation turn. 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., 2023; Muennighoff et al., 2025). Notably, this setting removes the explicit chain-of-thought (CoT) process, as recent studies have shown that explicit CoT provides no clear benefit for general machine translation (Wu et al., 2025a; Rajaee et al., 2026).

3.3 Sampling with a Selector

The 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 outputs for each source sentence, then directly report average translation performance in each sampling step.

3. **Sequential w/ selector**: we sequentially sample N outputs 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.3). We discuss other generation details in Appendix B.

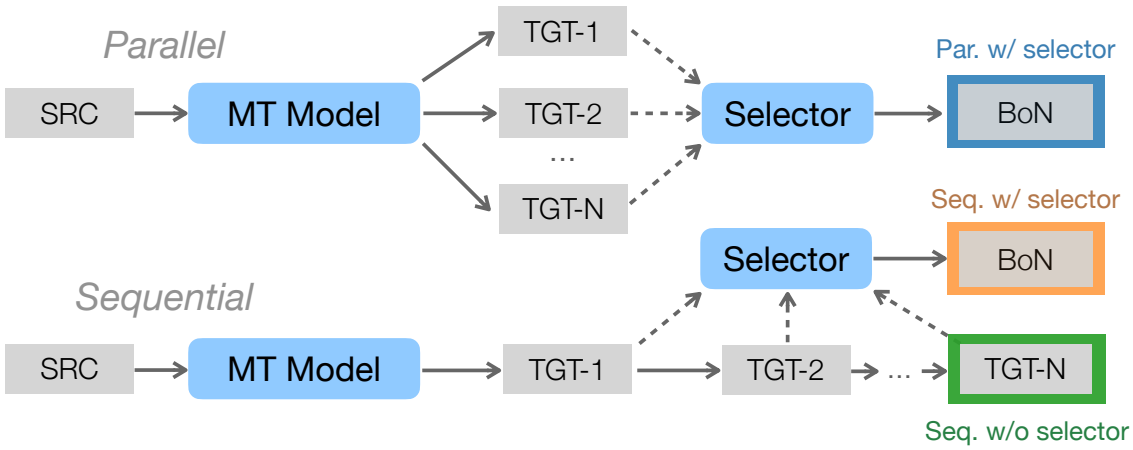


Figure 1: Illustration of the three sampling strategies investigated in this paper: 1) parallel sampling with selector, commonly referred to as Best-of-N sampling; 2) sequential sampling without a selector (sequential self-refinement), where we follow Wu et al. (2025a) to prompt model to *translate again for a better version*, and 3) sequential sampling with selector, which enables a fair comparison with parallel sampling (Best-of-N sampling).

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 mode) and `Qwen3-4B-Instruct-2507` (referred to as `Qwen3-4B`); Following Wu et al. (2025a), we also conduct experiments with a commercial LLM `GPT-4o-mini`.[4]

[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`
[4] gpt-4o-mini-2024-07-18

4.3 Metrics

Unless otherwise specified, we use CometKiwi-XL (Rei et al., 2023), 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 (i) MetricX-24 (Juraska et al., 2024), a hybrid metric used in the QE and reference-based modes, namely MetricX-24-Hybrid-XXL (qe) and MetricX-24-Hybrid-XXL (ref); (ii) a reference-based COMET (Rei et al., 2022); (iii) a reward model ReMedy-9B-22 (Tan and Monz, 2025), 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. Table 1 presented aggregated results for all 3 models, aggregated over the language directions. As described in §4.3, we use `CometKiwi-XL` as a selector. We summarize the key findings as follows.

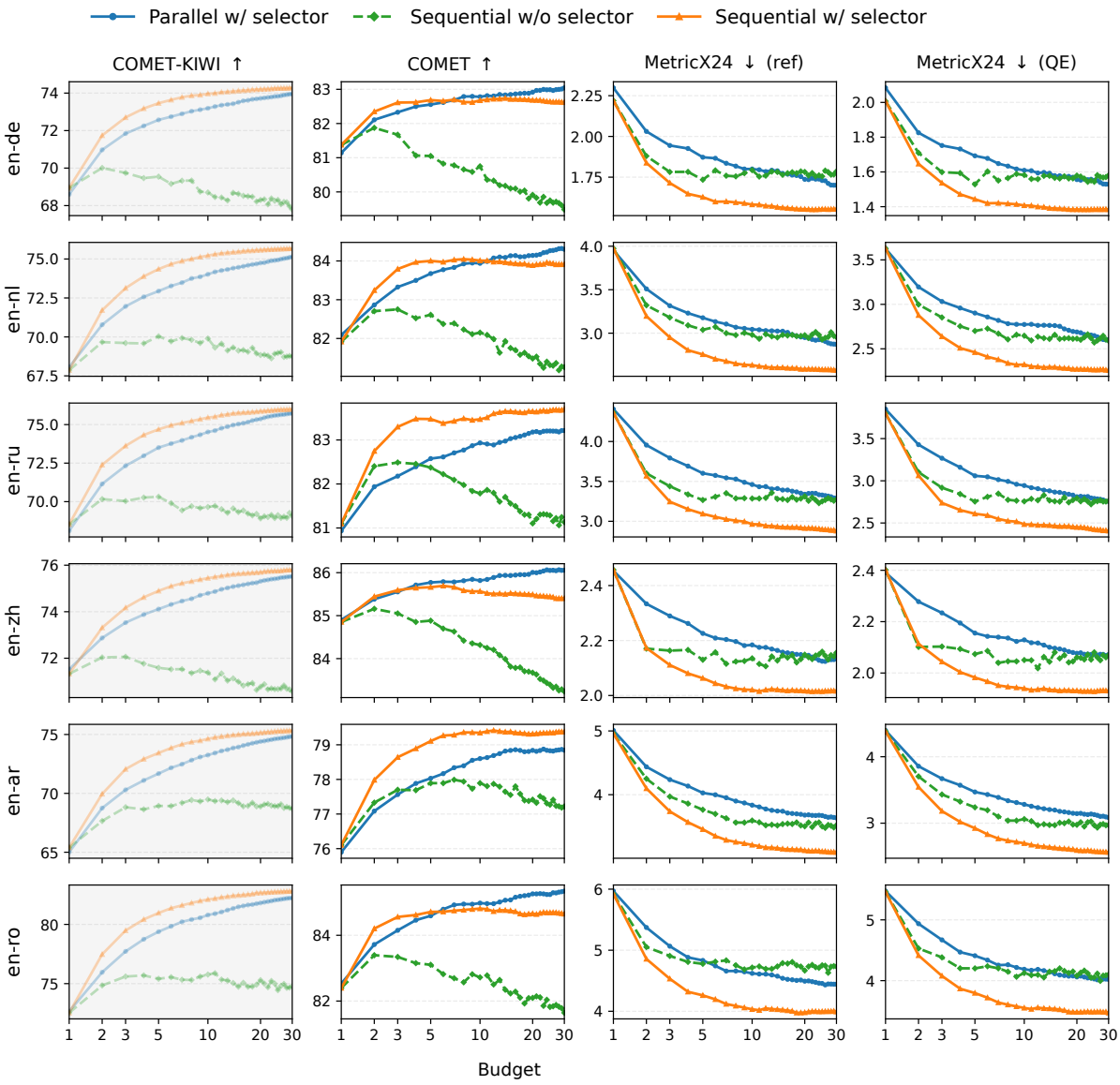


Figure 2: MT quality vs. the number of rounds of translation (budget) for the `Qwen-3-32B` model.

The self-refinement strategy works better in the early rounds. The green line in Figure 2 denotes the performance of self-refinement 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 left-a). These findings are also aligned with Wu et al. (2025a), where they report quality improvements for earlier rounds. Self-refinement patterns persist for commercial LLMs as well (Table 1, Figure 12). Recent LLMs research attributes these improvements to LLMs' reflection ability (Madaan et al., 2023; Muennighoff et al., 2025). In §6.2, we revisit this phenomenon and provide a more concrete explanation. When we evaluate the quality of sequential sampling for later rounds of translation (*e.g.*, $>$ 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 diversity.

Using 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.*, 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 CometKiwi-XL is used as both the selector and the evaluator,[5] shown here only to profile the Best-of-N sampling efficiency. Although the early iterations of self-refinement clearly improve average translation quality, the quality can degrade as the number of samples increases (at least according to COMET).[6]

By comparing results for sequential with and without selector (orange vs green), we observe that adding a selector results in 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 a selector. We thus treat sequential sampling without selector 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. Our main finding persists for commercial LLMs and for a smaller Qwen3 model (Table 1). 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 quality score, but 3 distinct MQM 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 §C 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), we notice a few critical improvements, most of which are attributed to sequential.

In Figure 3 left-(c), when we compare the selected samples from sequential vs parallel under higher budgets (30), we observe that sequential generates more fluent outputs with better style, but samples from parallel are more accurate. 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. Consistent with this hypothesis, our ablation results (§7) demonstrate that when the growing context is cut, sequential sampling recovers more stable quality behavior (according to COMET). More detailed annotation results are presented in Figure 15, revealing substantial cross-lingual variation across languages. In the English-to-Dutch direction, annotators judged the outputs to be of lower quality, whereas in the English-to-Chinese direction, annotators reported high quality of the outputs. These differences can plausibly be attributed to variations in model performance across languages. 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 Self-Refinement

As shown in Figure 2, sequential sampling is particularly effective in the early steps. A key question is what drives these improvements. Previous

[5] A setting most prone to metric hacking.

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

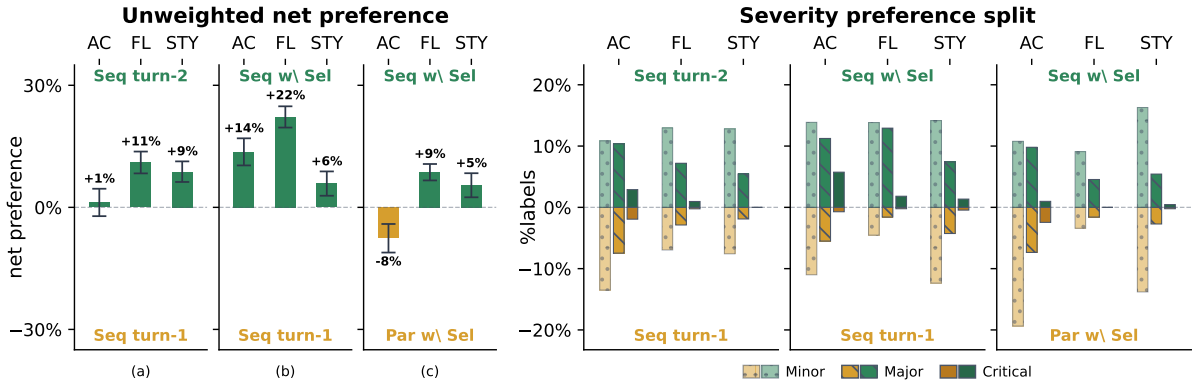
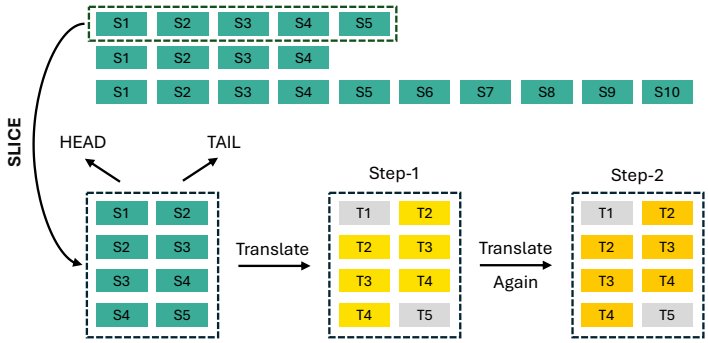


Figure 3: Human contrastive evaluation results: averaged preference split across three MQM quality dimensions: **accuracy** (AC), **fluency** (FL) and **style** (STY) for the `Qwen3-32B` model, averaged over 5 languages. (Left) shows the proportion of unweighted net preference (in %), and (right) presents preferences split across severity levels. Contrastive comparisons are (a) sequential turn-1 vs turn-2, (b) sequential with selector (budget-30) vs turn-1, (c) sequential vs parallel (with selector, budget-30). Error bars represent standard error of the mean (mean ± 1SE).



	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 self-refinement 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.

studies in the LLM community attribute this to the "reflection" abilities (Madaan et al., 2023; 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 self-refinement stems from the same effect: during first-pass generation, earlier and later segments differ 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,[7] ensuring strict comparability across positions. In practice, we use WMT24++ and maintain the sentence-to-document mapping using the metadata.

2. We then apply the sequential self-refinement strategy to the resulting corpus and compare

[7] 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.

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 §D.

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

6.3 Sequential Sampling is More Diverse for Smaller Budgets

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 context pool. We then 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 context and $D(y|H) = 1$ means the new output 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

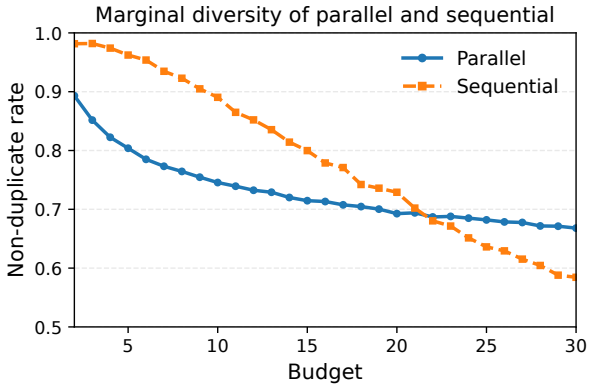


Figure 5: Diversity vs. budget for `Qwen-3-32B` model, avg. over 6 languages. Non-duplicate rate for each budget is computed *w.r.t.* previously sampled outputs, converging to 0 in the budget limit. Error bars, representing standard error of the mean ($\pm 1SE$), are nearly indistinguishable.

`Qwen-3-32B`, we run the experiments for a smaller model from the same family. Table 1 (with full results in Figure 11) presents the results for `Qwen-3-4B`. 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 self-refinement.

Larger sample context 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 context 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 1 results show that perhaps counterintuitively, larger context sizes do not imply better performance. The results for sequential with constrained context are better for almost every metric and language direction. Moreover, smaller context sizes h lead to less COMET degradation for the self-refinement mode (Figure 6, left). From

Model	Metric	Seq. w/o selector $t=1$	$t=2$	$t=3$	Seq. w/ selector $h=\infty$	$h=5$	$h=1$	Parallel
Qwen3 (32B)	COMET	81.33	82.11	82.13	83.28	83.27	83.72	83.47
	metricx24 (ref)	3.98	3.40	3.25	2.69	2.61	2.56	3.01
	metricx24 (qe)	3.61	3.05	2.90	2.34	2.27	2.22	2.68
Qwen3 (4B)	COMET	78.42	78.71	78.87	80.86	80.96	81.42	81.46
	metricx24 (ref)	5.69	5.04	4.85	4.07	4.02	3.84	4.16
	metricx24 (qe)	5.10	4.44	4.26	3.50	3.46	3.29	3.63
GPT (4o-mini)	COMET	82.22	82.86	82.93	83.96	84.01	84.23	83.81
	metricx24 (ref)	3.56	3.13	3.00	2.51	2.48	2.50	2.86
	metricx24 (qe)	3.29	2.88	2.74	2.25	2.21	2.24	2.60

Table 1: Automatic evaluation results for sequential and parallel sampling strategies aggregated across 6 languages. Left side presents results for self-refinement i.e. turn-$1, 2, 3$, of sequential sampling; right side shows the results with Best-of-N (budget 30) for the sequential (full context, $h = 5$, $h = 1$); and parallel with selector.

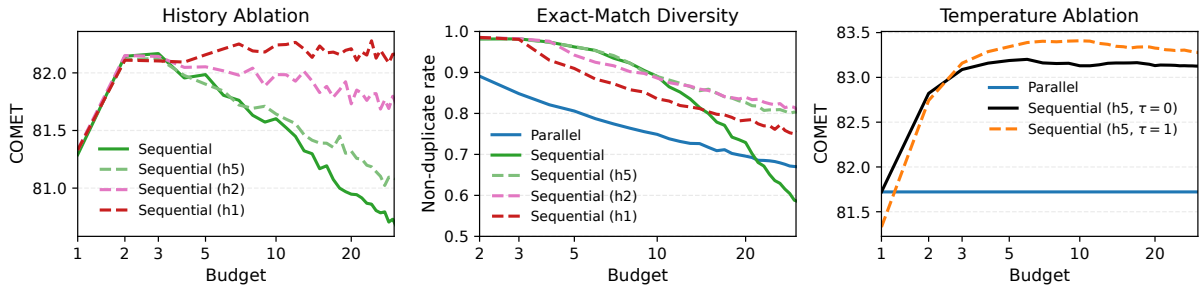


Figure 6: Ablation results for `Qwen-3-32B`. Left: equipping seq. w/o selector with smaller window size h results in less COMET degradation. Middle: context-controlled variants of sequential sampling show less diversity collapse. Right: sequential w/ selector sampling works even in the low-temperature limit, whereas parallel sampling collapses.

a sampling effectiveness angle, this result is particularly positive, as a smaller sequential context leads to both higher gains and lower costs. To explain this effect, we extend the diversity analysis (§6.3) to the context-size ablation experiment. From Figure 6 (middle), we observe that controlling the context effectively reduces the repetition bias, which explains the better results of sequential with a controlled context. 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. 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 6 (right) 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.

Additionally, 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.

Furthermore, sequential sampling may produce more fluent and natural translations, containing more minor inaccuracies, which can be risky in practical MT settings because fluent but unfaithful translations are hard for users to detect.

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. Our context size control analysis only suggests 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 context size, it is possible that the LLM has biases, steering its improvements away from certain aspects.

11 Acknowledgments

This work is supported in part by project VI.Veni.212.228 of the research program 'Veni', which is financed by the Dutch Research Council (NWO); this research is also funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers VI.C.192.080 and 2023.017; and is part of 'Hybrid Intelligence: augmenting human intellect' (https://hybrid-intelligence-centre.nl) with project number 024.004.022 of the research program 'Gravitation' which is (partly) financed by the Dutch Research Council (NWO).

We kindly thank Maya K. Nachesa, Annika Kniele and Wafaa Mohammed for their contributions to evaluation. We thank Zena Al-Khalili and members of the LTL lab for their helpful feedback.

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*.

Juraj Juraska, Daniel Deutsch, Mara Finkelstein, and Markus Freitag. 2024. MetricX-24: The Google submission to the WMT 2024 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, pages 492–504, Miami, Florida, USA. Association for Computational Linguistics.

Ryo Kamoi, Yusen Zhang, Nan Zhang, Jiawei Han, and Rui Zhang. 2024. When can LLMs actually correct their own mistakes? a critical survey of self-correction of LLMs. *Transactions of the Association for Computational Linguistics*, 12:1417–1440.

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, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*.

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, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

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 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.

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, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

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 Tomz, Christopher D Manning, and Weiyan Shi. 2026. Verbalized sampling: How to mitigate mode collapse and unlock LLM diversity. In *Forty-third International Conference on Machine Learning*.

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 §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 §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` on 4 `NVIDIA RTX A6000` cards with up to 8 request processing at the same time. With the total budget 30, each full-context sequential run, for a single language pair takes on average 14h; each context-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. Context-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 2.

Regarding the KV-cache friendliness, full-context sequential sampling can naturally reuse the previous cached activations, whereas context-k sampling requires cache recomputation, with context-k requiring recomputing only one previous outputs. Parallel sampling enjoys reusing the KV-cache of the shared prefixes (*e.g.* prompt).

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 2: `Qwen3-*` generation parameters.

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 sometimes 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.

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 21), which highlights this phenomenon.

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(§C, 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 §C, and can optionally leave the notes, explaining their decisions. We include a screenshot of our interface in Figure 8. We present the human annotation instructions in Figure 7.

D Experimental Details for §6.2

We conduct our experiments on `Qwen-3-32B` model. The self-refinement 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[8] 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 9, 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.

F Results

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

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

Qwen3-4b-instruct additional results. Results for the `Qwen3-4b-instruct` are presented in Figure 11.

GPT-4o-mini additional results. Results with full-context sequential sampling for the `GPT-4o-mini` are presented in Figure 12. The context-5 ablation is presented in Figure 13, and the context-1 ablation is presented in Figure 14.

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

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

```
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.
```

Figure 7: Human annotation instructions.

sequential vs parallel under budget 30, we notice that parallel often demonstrates higher accuracy.

H Extended Ablation Results

H.1 Context size ablation

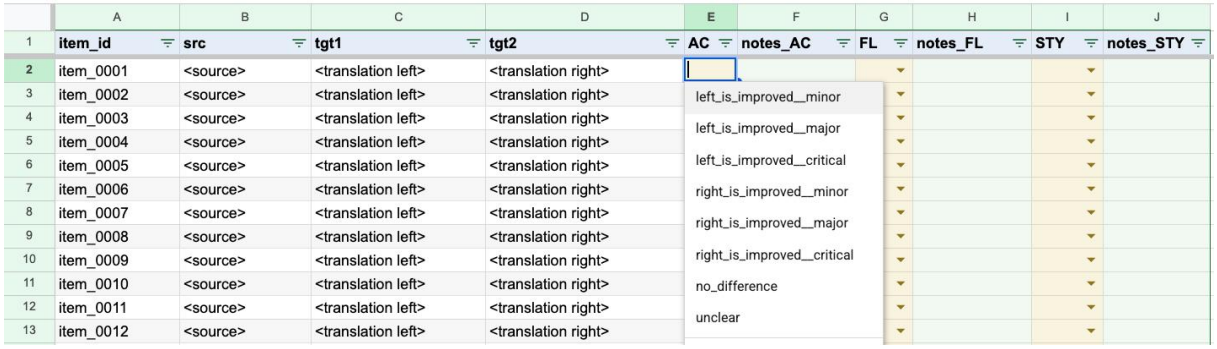
In this section, we present the results of context managing for sequential sampling using the sliding window approach with context sizes: full context, context-5, context-2, context-1. Results without selector show substantial quality improvements in Figure 18, when constraining the context. Figure 19 presents results for sequential sampling with selector, where we observe roughly the same pattern, but with a smaller difference between runs, smoothed by a selector.

H.2 Results with another selector metric

Figure 20 presents a version of main results, using MetricX24 (qe) as a selector instead of `COMET-KIWI`. Results under MetricX24 (qe) as selector are consistent with the main results regarding our main finding: sequential sampling explores the output spaces more effectively, compared to parallel sampling, especially under a limited budget.

H.3 Sampling hyperparameters

We use temperature 1.0 for all experiments, unless stated otherwise. Figure 21 visualizes quality vs budget scaling using Best-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;



	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 8: 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.

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 context experiments, where we set it to 764 to prevent model length overflows (given a small number of collapsed extra long outputs).

		Seq. w/o selector			Seq. w/ selector			Parallel
Lang	Metric	$t = 1$	$t = 2$	$t = 3$	$h = \infty$	$h = 5$	$h = 1$	
en-de	COMET	81.26	81.59	81.49	82.62	82.24	82.84	**83.04**
	metricx24 (ref)	2.23	1.92	1.81	1.55	1.52	**1.46**	1.70
	metricx24 (qe)	2.02	1.72	1.62	1.38	1.36	**1.32**	1.53
en-nl	COMET	82.10	82.81	82.70	83.92	84.02	**84.52**	84.30
	metricx24 (ref)	3.94	3.33	3.16	2.57	2.47	**2.42**	2.87
	metricx24 (qe)	3.63	3.00	2.85	2.26	2.18	**2.12**	2.60
en-ru	COMET	80.98	82.44	82.35	83.70	83.51	**84.00**	83.21
	metricx24 (ref)	4.41	3.60	3.45	2.88	2.82	**2.73**	3.29
	metricx24 (qe)	3.86	3.09	2.96	2.41	2.31	**2.22**	2.76
en-zh	COMET	84.85	85.14	85.08	85.40	85.59	85.79	**86.06**
	metricx24 (ref)	2.47	2.20	2.21	2.02	1.98	**1.92**	2.13
	metricx24 (qe)	2.39	2.14	2.14	1.93	1.90	**1.85**	2.07
en-ar	COMET	76.11	77.30	77.54	79.37	79.40	**79.69**	78.84
	metricx24 (ref)	4.92	4.25	3.99	3.10	**3.01**	3.03	3.64
	metricx24 (qe)	4.37	3.72	3.44	2.56	**2.49**	2.52	3.08
en-ro	COMET	82.68	83.41	83.60	84.65	84.84	**85.46**	85.34
	metricx24 (ref)	5.90	5.10	4.88	3.99	3.87	**3.80**	4.44
	metricx24 (qe)	5.39	4.65	4.37	3.47	3.40	**3.30**	4.02

Table 3: Automatic evaluation results for sequential and parallel sampling strategies across 6 languages for the `Qwen3-32B` 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 vs turn2				turn1 vs seq. w/ sel				par. w/ sel vs 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 4: 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 only partially finished for en-de due to time constraints.

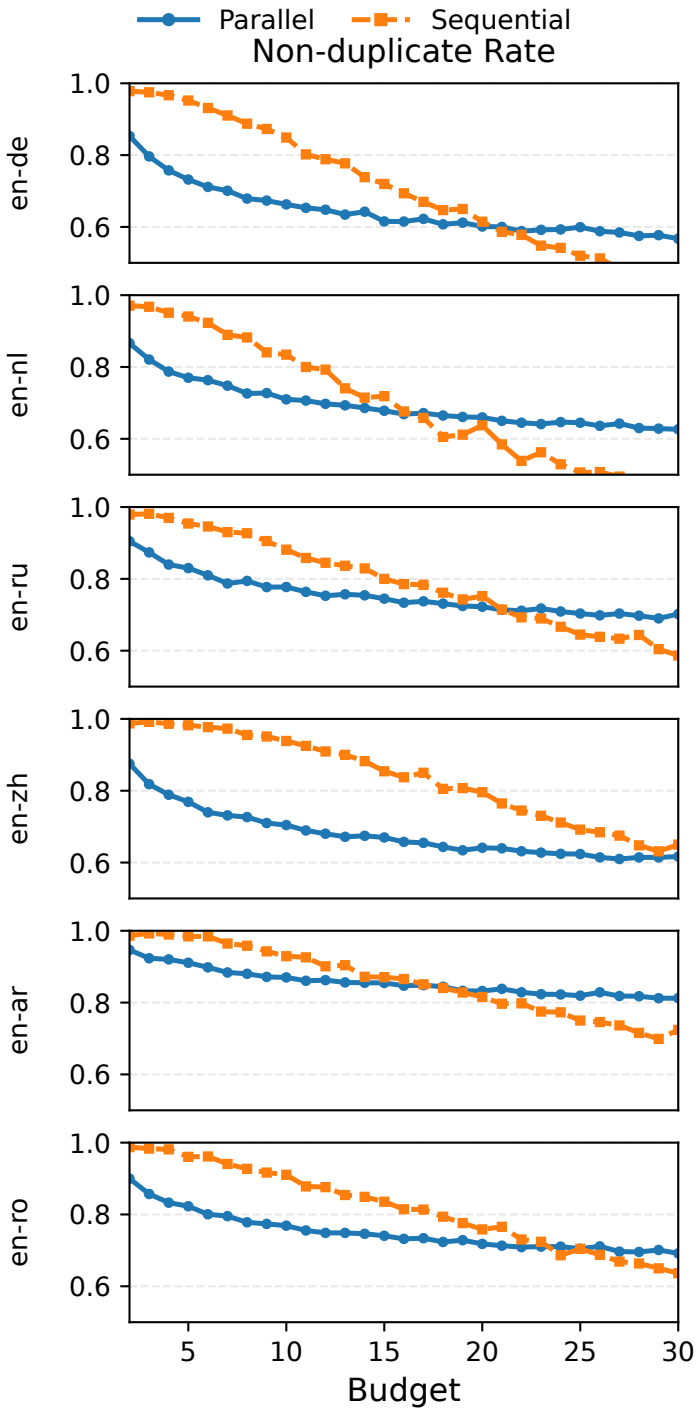


Figure 9: Non-duplicate rate for `Qwen-3-32B` model for parallel and sequential sampling per language. Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

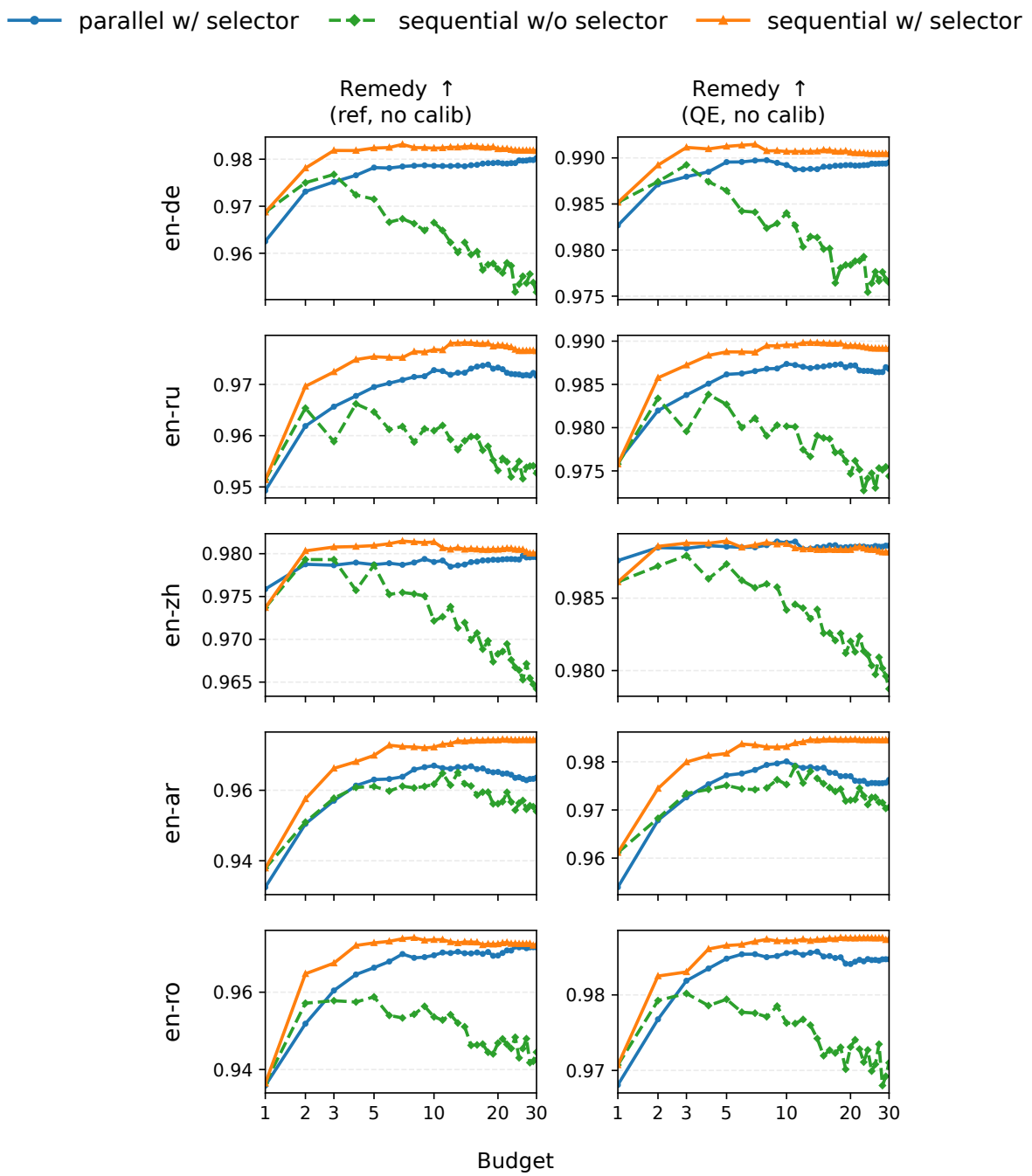


Figure 10: Additional evaluation with the ReMedy (`ReMedy-9B-22`) metric (Tan and Monz, 2025) for 5 language directions (en-nl is not supported by the metric). Samples are from the `Qwen-3-32B` model. Quality vs. budget. Selector is `COMET-KIWI`.

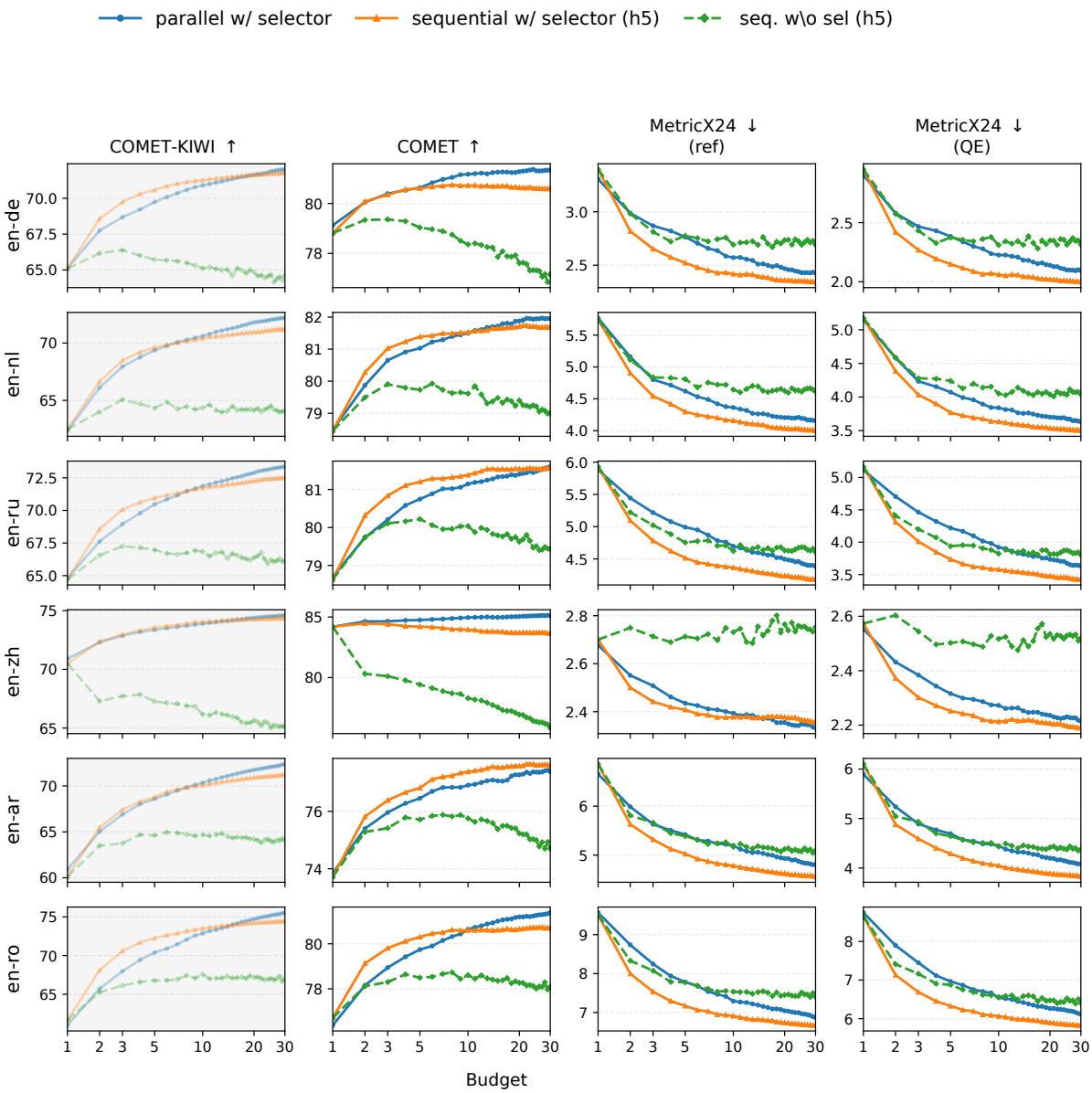


Figure 11: Quality vs budget results for the `Qwen-3-4B-2507` model. Selector is `COMET-KIWI`.

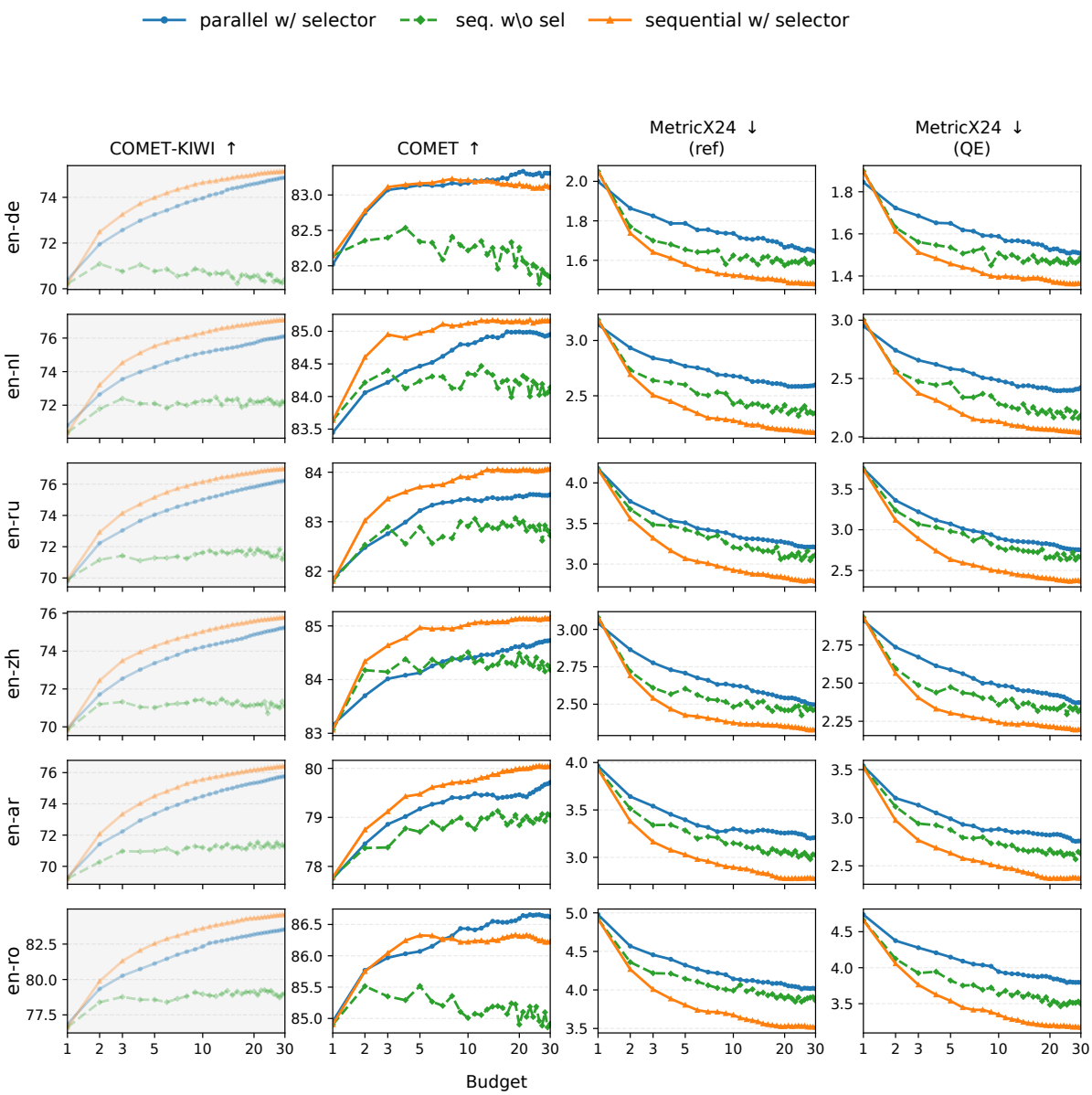


Figure 12: Quality vs budget results for the `GPT-4o-mini` model. Selector is `COMET-KIWI`.

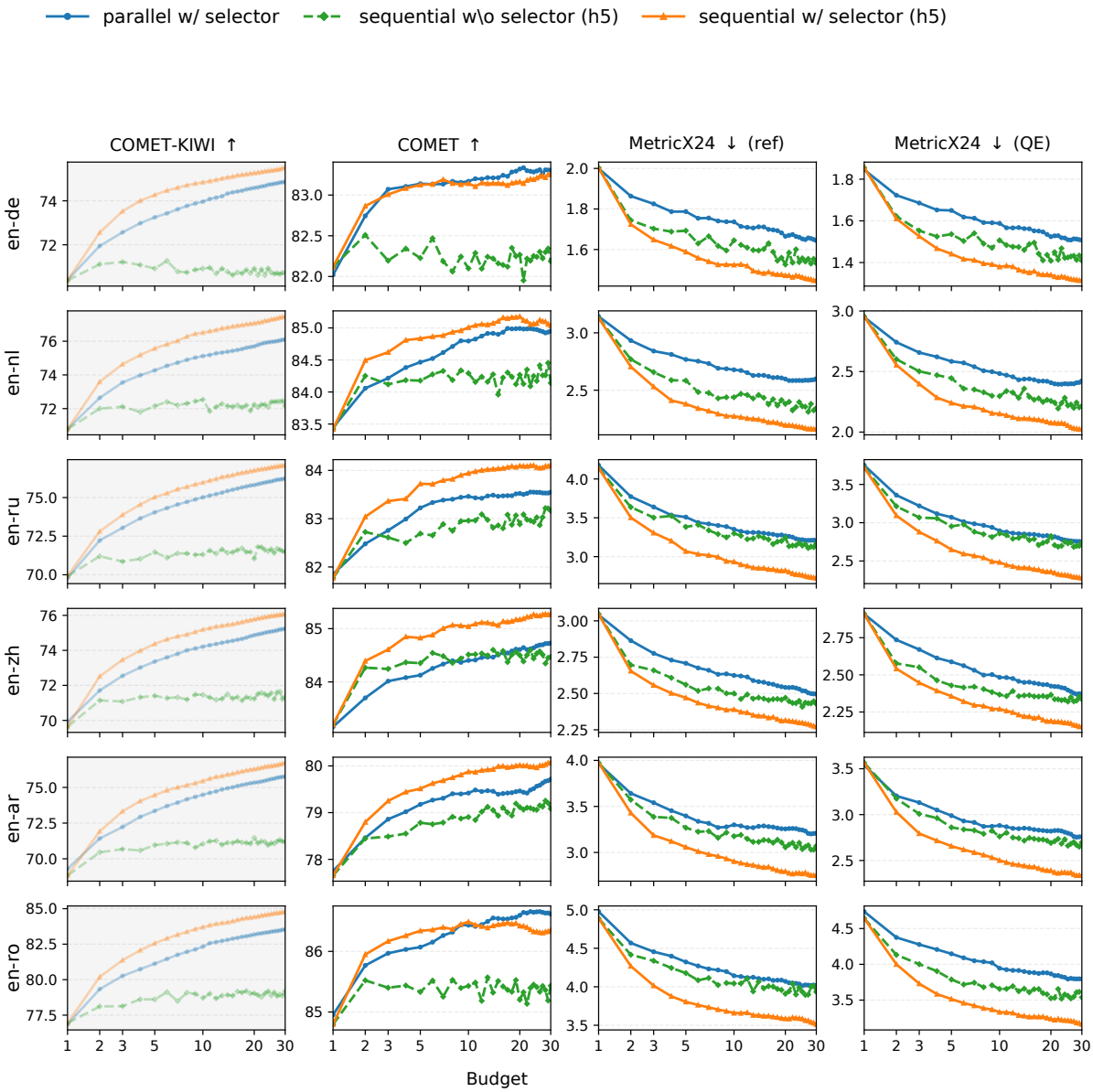


Figure 13: Quality vs budget results for the `GPT-4o-mini` model. Sequential sampling with context-5 ablation. Selector is `COMET-KIWI`.

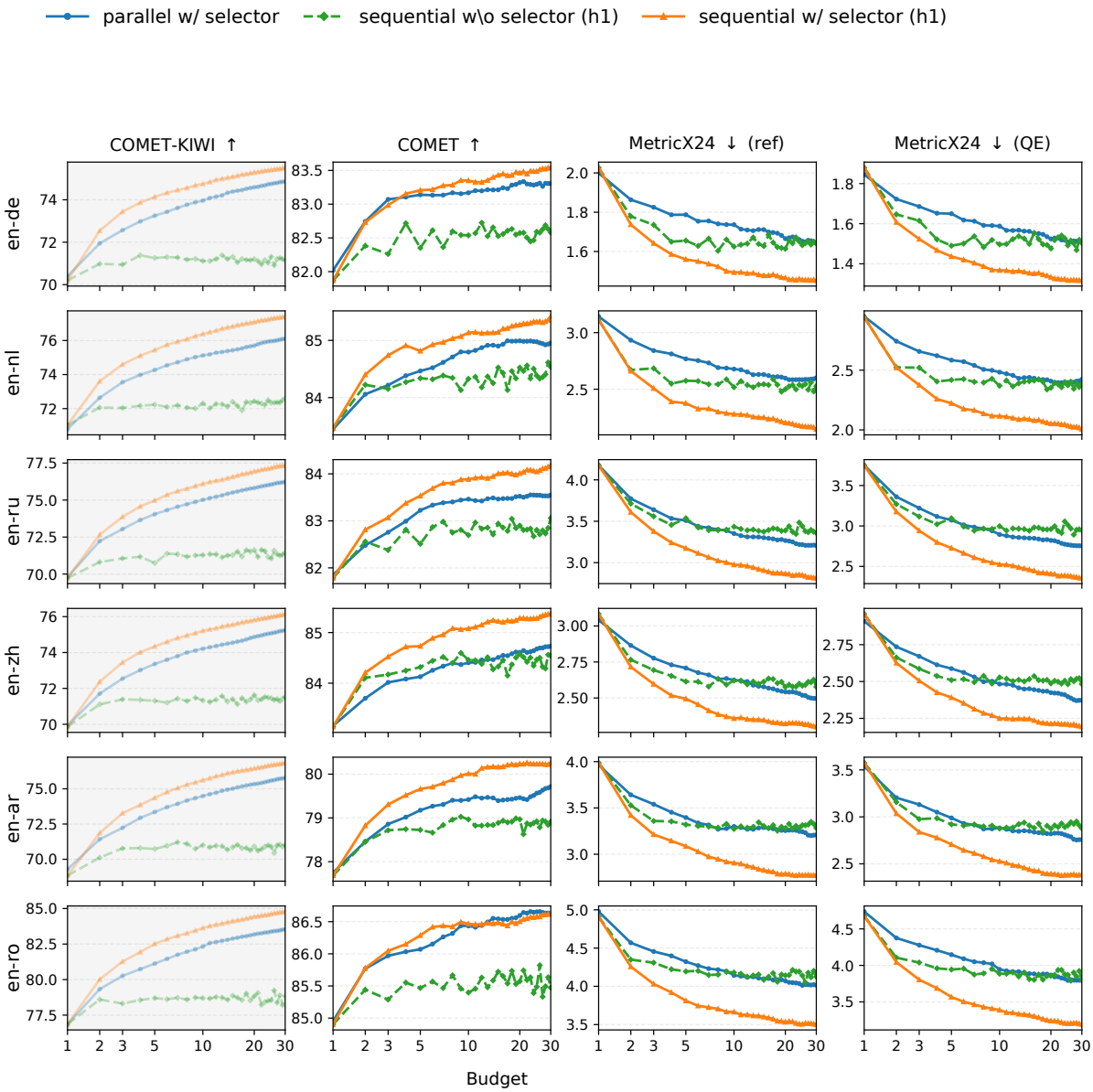


Figure 14: Quality vs budget results for the `GPT-4o-mini` model. Sequential sampling with context-1 ablation. Selector is `COMET-KIWI`.

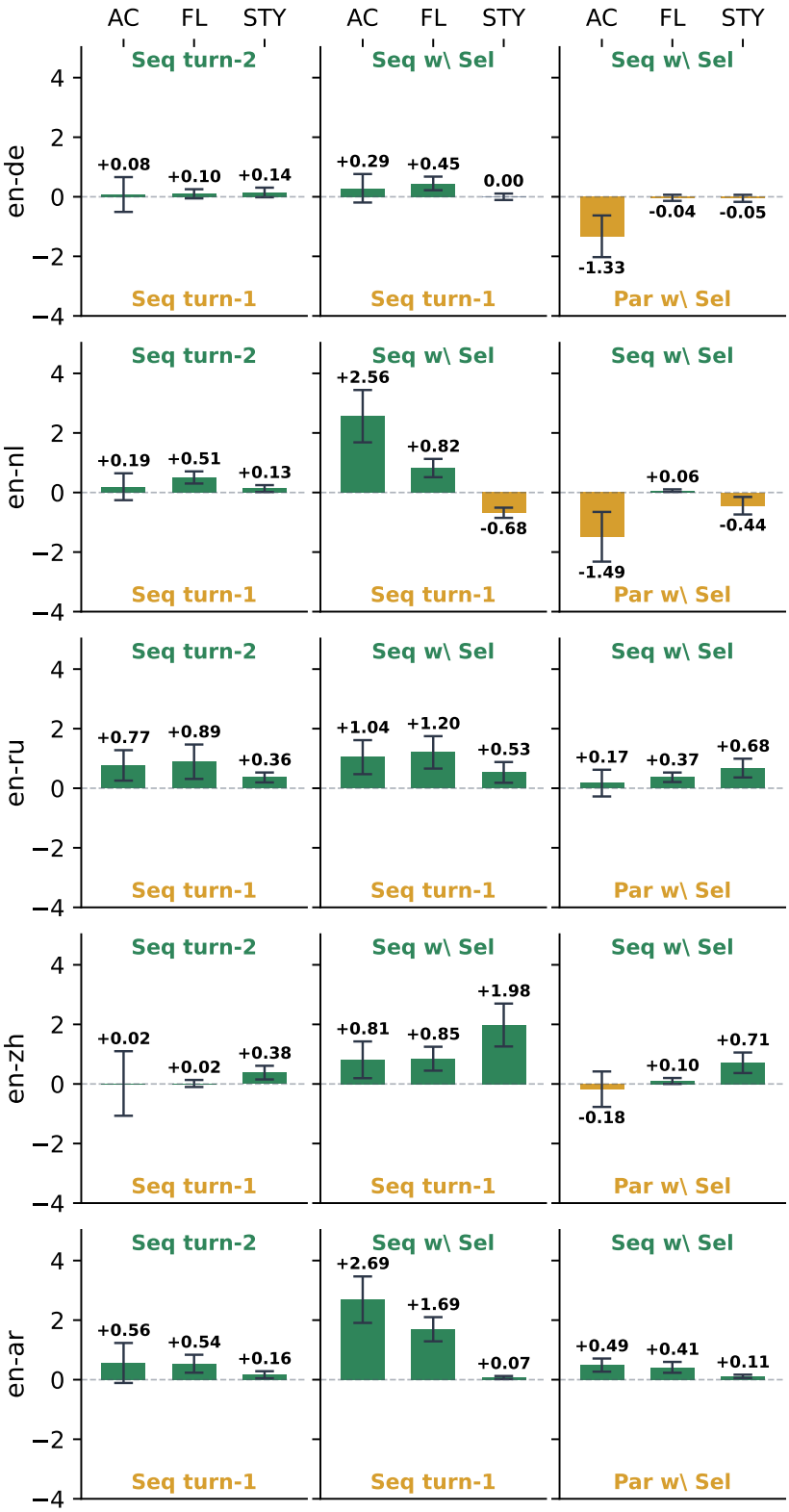


Figure 15: Human annotation results using MQM **average weighted relative improvement** for accuracy (AC), fluency (FL) and style (STY) for the for the `Qwen3-32B` model. Results are presented for (i) turn-1 vs turn-2, (ii) Sequential w Sel (budget 30) vs Seq. turn-1, (iii) Sequential w Sel (budget 30) vs Parallel w Sel (budget 30). Error bars represent standard error of the mean.

Figure 16: Diversity and MT quality (COMET) for `Qwen-3-32B` model for parallel sampling. Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

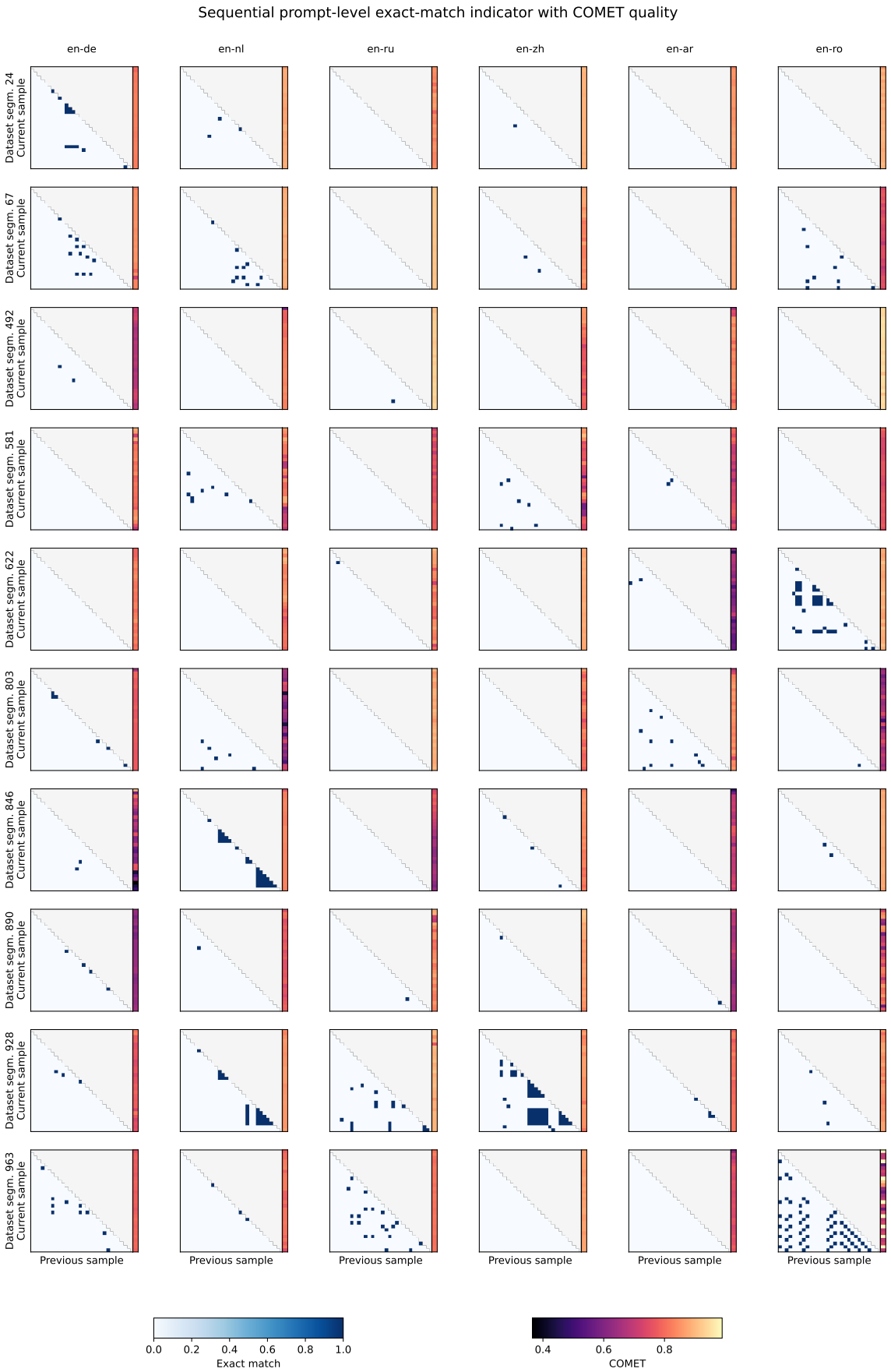


Figure 17: Diversity and MT quality (COMET) for `Qwen-3-32B` model for sequential sampling. Diversity metric represents how close a new sample is to every previous sample using exact match similarity.

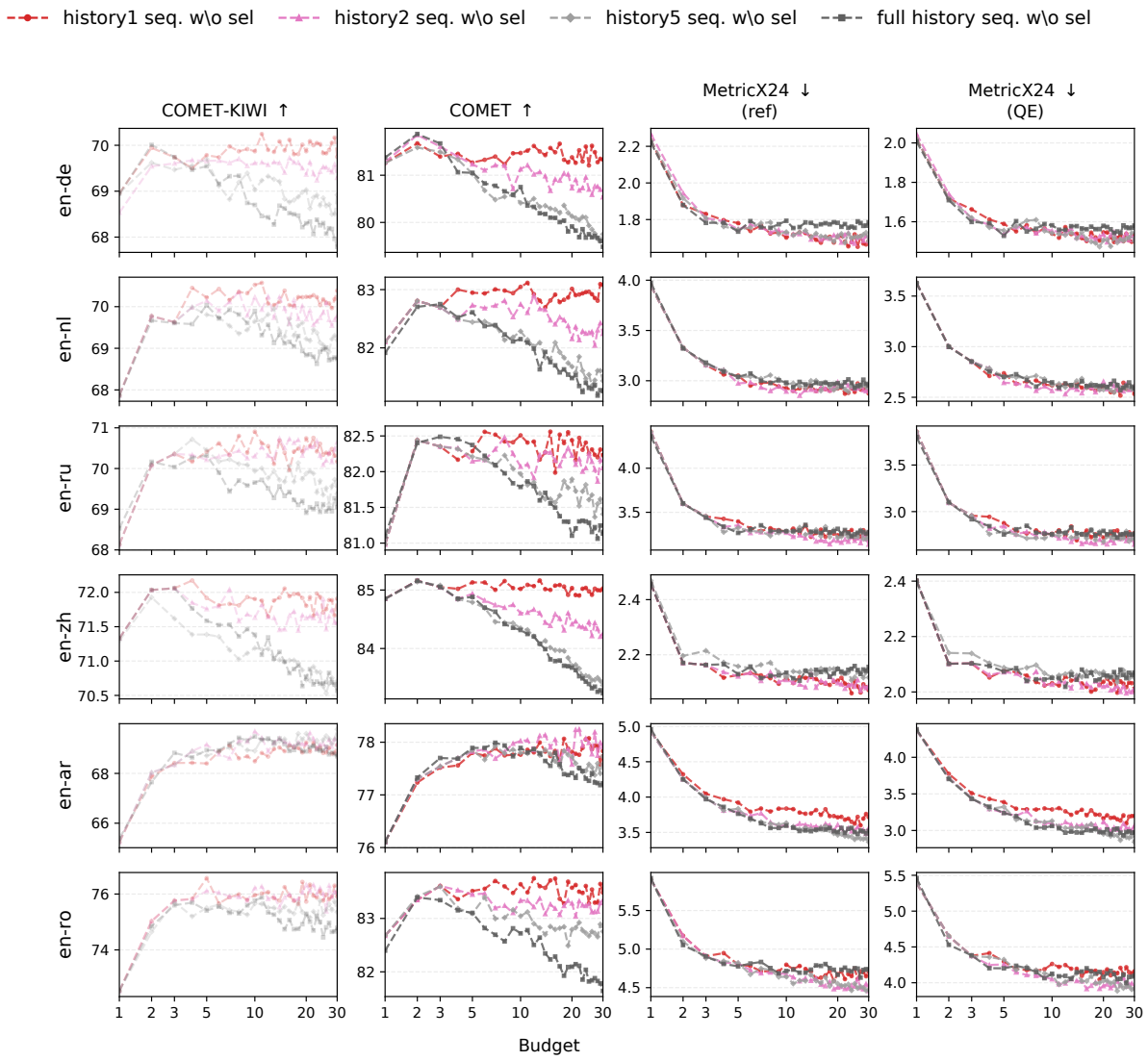


Figure 18: Context size ablation for the sliding window sequential sampling approach for the `Qwen-3-32B` model. The scaling of self-refinement (*i.e.* **sequential sampling without selector**) can be largely improved by compacting the context using the sliding window approach.

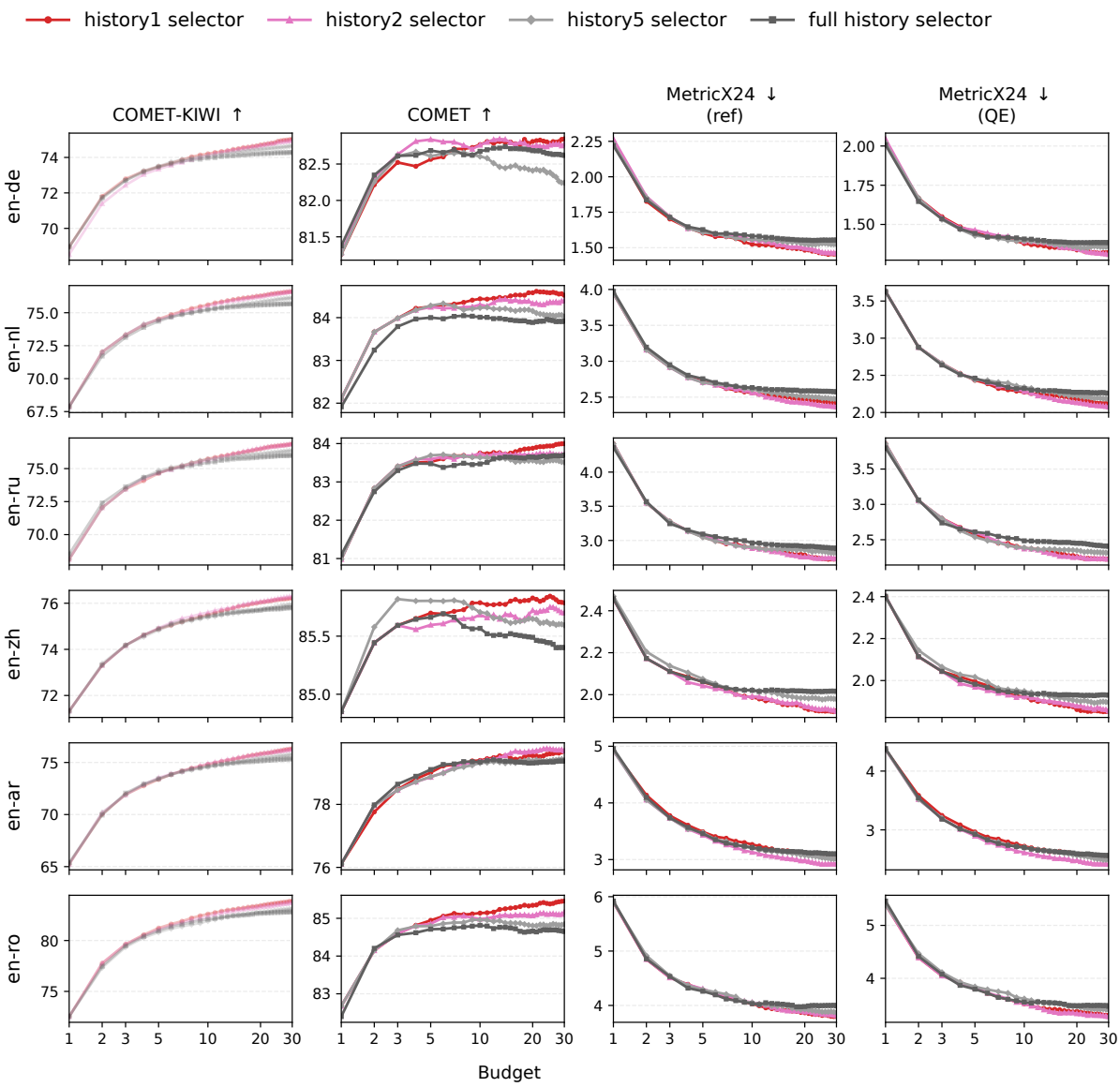


Figure 19: Context size ablation for the sliding window sequential sampling approach for the `Qwen-3-32B` model. The quality of **sequential sampling with selector** can be slightly improved by compacting the context using the sliding window approach.

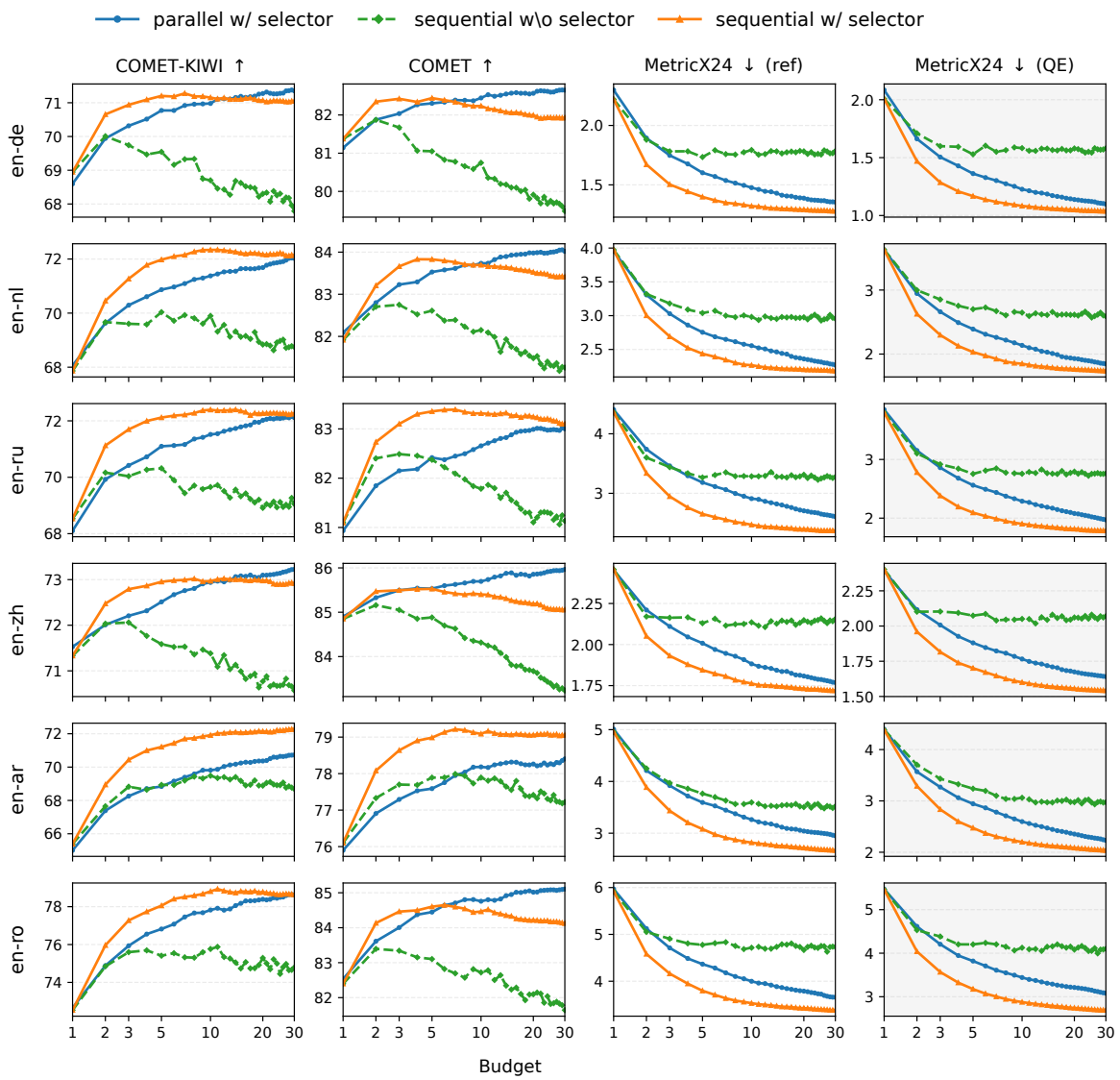


Figure 20: Quality vs. budget for the `Qwen-3-32B` model, when using a different selector model: MetricX24 (qe). Budget denotes the number of rounds of translation for sequential or parallel.

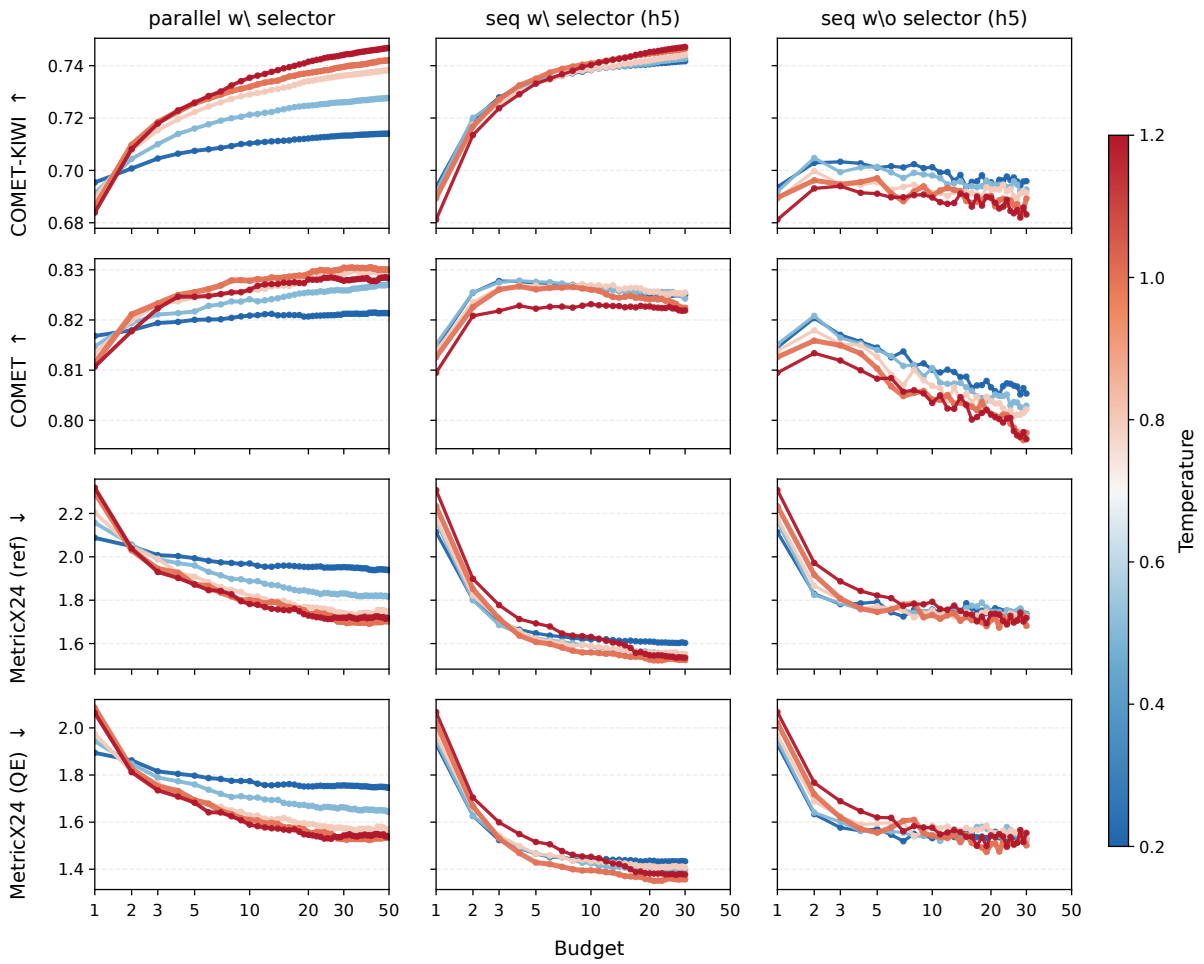


Figure 21: Temperature ablation for parallel and sequential runs for the en-de language pair. Temperature 1.0 is near optimal for the Best-of-N sampling results. For the self-refinement (no selector), lower temperature values lead to better results for the MT evaluation (high-precision, but low-diversity regime).