

Self-Evolving Optimization of Agentic Systems for Machine Translation

Zhan Shen¹ Xiaotian Wang¹ Jason Naradowsky¹ Yusuke Miyao^{1,2}

¹Department of Computer Science, The University of Tokyo

²Research and Development Center for Large Language Models,
National Institute of Informatics

shenzhan@ecc.u-tokyo.ac.jp, xtwang@is.s.u-tokyo.ac.jp

jason.narad@gmail.com, yusuke@is.s.u-tokyo.ac.jp

Abstract

LLM-based translation agents can plan, review, and revise translations, but most existing systems still rely on a single hand-crafted instruction set shared across inputs. This is suboptimal for translation, where texts vary widely in domain, context, style, and cultural grounding, leading to heterogeneous and input-specific errors. We propose STEINS, a self-evolving framework that treats the instruction set governing agent roles, translation strategies, and interaction constraints as an optimization target. During optimization, STEINS performs structured error diagnosis and uses the resulting evidence to refine the agent specification in a targeted and interpretable manner. The framework supports both offline corpus-level evolution and online instance-level adaptation at test time. Across WMT23 (general-domain, segment-level) and LITEVAL-CORPUS (literary, paragraph-level), and across four agentic translation architectures, offline evolution yields modest but generally positive corpus-level gains, while online instance-level adaptation yields larger and more consistent improvements, with macro-averaged XCOMET gains of +1.28 to +3.49 over 12 settings regardless of whether adaptation starts from the original or offline-evolved specification. Overall, our findings identify instance-level specification adaptation as the key ingredient in effective agentic machine translation.

1 Introduction

Large language models (LLMs) have enabled a new generation of agentic machine translation systems, ranging from single-agent pipelines (Wang et al., 2024b) to multi-agent collaborative frameworks (Dutta et al., 2025; Wu et al., 2024). In these systems, translation quality depends critically on the *agent specification*: the instructions that define agent roles, translation strategies, and interaction rules. In most existing systems, however, this specification is manually designed and then

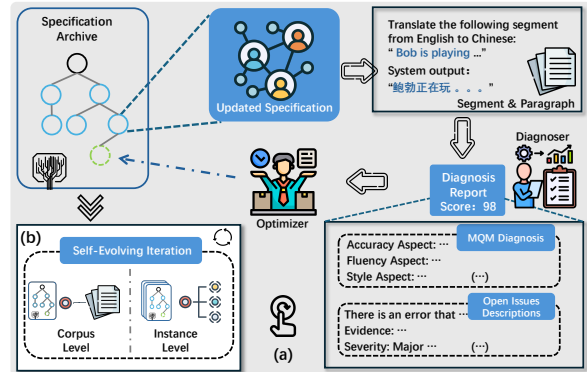


Figure 1: Schematic illustration of the STEINS framework. (a) The general diagnose-and-optimize loop. (b) The framework is instantiated for both offline corpus-level evolution and online instance-level adaptation.

kept fixed after deployment, forcing a single shared instruction set to handle inputs that differ widely in domain, context, style, and cultural grounding.

We reframe this problem through two complementary hypotheses: (1) hand-crafted agent specifications are often suboptimal and can be improved through data-driven refinement; and (2) the optimal specification is partly input-dependent: translation errors are heterogeneous and context-sensitive, so a specification that is adequate on average may still be poorly matched to a particular input. These point to two distinct levels of agent optimization: corpus-level specification evolution and instance-level adaptation at test time.

Existing work on automated instruction refinement addresses this problem only partially. Most methods optimize prompts or workflows at the dataset level, provide only coarse global feedback for targeted updates, and rarely support instance-level adaptation at test time. Generate-and-select methods such as APE (Zhou et al., 2022) and OPRO (Yang et al., 2023) rank candidate prompts on held-out data, while evolutionary methods such as EvoPrompt (Guo et al., 2023) and Prompt-

Breeder (Fernando et al., 2023) apply generic search operators over prompt populations. While effective as general-purpose optimization strategies, these approaches are not designed for machine translation and provide limited translation-specific diagnostic signal for targeted refinement.

In this work, we propose STEINS (Self-Evolving Translation Enhancement In AgeNtic Systems)¹, a self-evolving framework that shifts specification optimization from generate-and-select to diagnose-and-optimize. During specification optimization, an LLM-based diagnoser performs structured Multidimensional Quality Metrics (MQM)-grounded error analysis and uses the resulting evidence to refine the agent specification in a targeted and interpretable manner. STEINS supports both offline corpus-level evolution and online instance-level adaptation, and can be applied to multiple agentic translation architectures without modifying the underlying translation model. To support broader search over the specification archive, we introduce a temperature-controlled softmax mechanism that biases rewrite selection toward higher-credit historical specifications, enabling branching in the specification history. Across two benchmarks, three language pairs, and four agentic translation architectures, our experiments reveal a clear offline-online asymmetry in role and payoff. Offline evolution yields modest but generally positive corpus-level gains and provides an efficient initialization, whereas the largest and most consistent improvements come from online instance-level adaptation. In summary, our contributions are as follows:

- We propose STEINS, a diagnosis-driven framework for specification optimization in agentic machine translation, which performs structured, span-grounded error diagnosis and turns the resulting evidence into targeted, interpretable updates to the agent specification.
- We unify offline corpus-level evolution and online instance-level adaptation in a single self-evolving framework, enabling controlled comparisons and analysis of their distinct behaviors and trade-offs in agentic translation.
- We show that instance-level adaptation delivers the largest gains in agentic translation, while offline evolution provides a modest but useful corpus-level improvement

and an efficient starting point for later test-time adaptation. Across WMT23 (Kocmi et al., 2023) and LITEVAL-CORPUS (Zhang et al., 2025b), online adaptation yields macro-averaged XCOMET gains of +1.28 to +3.49 over 12 settings.

2 Related Work

Automated Prompt and Agent Optimization

Autonomous self-improvement, long discussed as a theoretical possibility in computation (Good, 1966; Schmidhuber, 1987, 2007), has recently become a practical paradigm in LLM-based systems, where prompts, instructions, workflows, and even agentic designs can be iteratively improved from feedback. Prompt-level methods such as APE (Zhou et al., 2022), OPRO (Yang et al., 2023), Prompt-Breeder (Fernando et al., 2023), EvoPrompt (Guo et al., 2023), and TextGrad (Yuksekgonul et al., 2024) iteratively propose and refine textual instructions or variables using search, feedback, or evolutionary mechanisms. More recent work extends this perspective from prompts to agent- or system-level design, including ADAS (Hu et al., 2024), Gödel Agent (Yin et al., 2024), AlphaEvolve (Novikov et al., 2025), and the Darwin Gödel Machine (Zhang et al., 2025a). However, these approaches are largely general-purpose and typically optimize shared prompts, workflows, or agent designs at the task level through global search, scalar feedback, or generic mutation operators. They are not designed for machine translation and provide limited translation-specific diagnostic signal for targeted, per-input refinement of agent specifications.

Adaptive and Agentic Machine Translation

Recent work improves LLM-based translation through structured collaboration, memory, retrieval, reflection, and learning-based adaptation beyond static zero-shot pipelines. Representative agentic frameworks include role-specialized collaboration (Wu et al., 2024), memory-augmented document translation (Wang et al., 2024b), and retrieval- and reflection-enhanced translation (Chen et al., 2024). Related work also explores reflective or deliberative translation, from iterative self-refinement frameworks such as TEaR (Feng et al., 2024) to debate-based and long-reasoning approaches (Liang et al., 2023; Wang et al., 2024a). A further line improves translation through model-level adaptation, as in R1-T1 (He et al., 2025) and MT-R1-Zero (Feng et al., 2025), which optimize translation

¹<https://github.com/ZhanShenYo/STEINS>

behavior via reward-guided training. In contrast, STEINS treats the agent specification itself as the adaptation target. Rather than proposing another fixed workflow or updating model parameters, it refines the instructions governing agent behavior through evidence-grounded diagnosis, and explicitly studies both offline corpus-level evolution and online instance-level adaptation.

3 The STEINS Framework

We cast agentic machine translation as search over the space of *agent specifications* $\mathcal{S}$. Depending on the architecture, a specification $s \in \mathcal{S}$ may be instantiated as a single prompt or as a coordinated set of prompts. For a source input x , the translation system $\mathcal{T}$ produces a translation $\hat{y} = \mathcal{T}(x; s)$ and a process log ℓ that records intermediate agent outputs or interactions.

3.1 Problem Setup

Let $D = \{(x_j, y_j)\}_{j=1}^{|D|}$ denote a translation corpus with reference translations. In offline corpus-level evolution, updates are performed on a training split with reference translations, and the goal is to learn a reusable corpus-level specification

$$s^* = \arg \max_{s \in \mathcal{S}} \mathbb{E}_{(x,y) \sim D} [m(\mathcal{T}(x; s), y)],$$

where $m(\cdot, \cdot)$ is a reference-based MT metric. In online instance-level adaptation, by contrast, adaptation is performed directly on test instances, with reference translations neither exposed to nor used by the adaptation policy. For each input x , we optimize over specifications reachable within at most R adaptation rounds:

$$s_x^* = \arg \max_{s \in \mathcal{S}_x(R)} m_{\text{qe}}(\mathcal{T}(x; s)),$$

where $m_{\text{qe}}(\cdot)$ is a reference-free quality estimator, and $\mathcal{S}_x(R)$ denotes the corresponding reachable specification set.

STEINS approximates both objectives with a shared *translate-diagnose-optimize* loop over a specification archive. Each execution is evaluated using the metric appropriate to the current phase: the reference-based m during offline evolution and the reference-free m_{qe} during online adaptation. The resulting quality signal is assigned to the newly created child as search credit for subsequent rewrite selection. Because test-time search is not necessarily monotonic across rounds, the final online prediction is chosen as the translation with the highest

round-level quality under m_{qe} , rather than the last-round output. In our experiments, we instantiate m with XCOMET (Guerreiro et al., 2024) and m_{qe} with XCOMET-QE.

3.2 Diagnose-Optimize Operator

The core component in STEINS is a *diagnose-optimize* operator that uses diagnostic evidence to transform a selected specification s into an updated specification s' . The diagnoser performs MQM-style (Freitag et al., 2021) error analysis over the current translation, using the source text and translation output as primary inputs and the process log as auxiliary context; additional phase-specific inputs are described in Sections 3.4 and 3.5. Each issue is labeled with an error category and severity, and grounded in explicit source-side and target-side spans, yielding structured and verifiable diagnostic evidence. MQM definitions are provided in Appendix 20. The optimizer applies the resulting diagnostic evidence to an editable *strategy layer*, while *protected regions* containing fixed task instructions, placeholders, and output constraints remain unchanged. This preserves the task interface while enabling targeted and interpretable strategy updates.

3.3 Search Archive and Rewrite Selection

The generated specifications form a tree-structured search history, represented conceptually as

$$\mathcal{A} = \{(s_i, p_i, f_i)\}_{i=0}^N,$$

where s_i is a specification, p_i is its parent index, and f_i is its search credit. The root s_0 is the starting specification for the current search, with $p_0 = \perp$ and $f_0 = 0$. In offline evolution, s_0 is the original hand-crafted specification; in online adaptation, it is either the original specification or the offline-evolved specification, depending on the initialization setting. At each update step, the rewrite-selection policy chooses an archived specification to revise in light of the current diagnostic evidence. The resulting child is appended to $\mathcal{A}$, with the current quality signal assigned as its search credit.

Sequential selection. Select the most recently added node, yielding a single-branch trajectory.

Softmax sampling. Let $\mathcal{C}_t$ denote the eligible archive nodes at step t . For $i \in \mathcal{C}_t$,

$$P_t(i) = \frac{\exp(f_i/\tau - \beta \ln(1 + c_i))}{\sum_{j \in \mathcal{C}_t} \exp(f_j/\tau - \beta \ln(1 + c_j))}.$$

Algorithm 1 Shared search skeleton of STEINS

Require: phase ϕ , input z , starting specification s_0 , translation system $\mathcal{T}$, rewrite-selection policy π , budget B
Ensure: phase-dependent output

```

1:  $\mathcal{A} \leftarrow \{(s_0, \perp, 0)\}; \mathcal{H} \leftarrow \emptyset$ 
2: for  $t = 1$  to  $B$  do
3:    $e_t \leftarrow t - 1$ ; if  $\phi = \text{OFFLINE}$  then  $e_t \leftarrow$ 
     SELECTREWRITE $_{\phi}(\mathcal{A}, \pi)$ 
4:    $\hat{y}_t, \ell_t \leftarrow \mathcal{T}(z; s_{e_t})$ 
5:    $q_t \leftarrow \text{EVALUATE}_{\phi}(z, \hat{y}_t)$ ; APPEND( $\mathcal{H}, (t, \hat{y}_t, q_t)$ )
6:   if  $\phi = \text{ONLINE}$  and  $(\text{CONVERGED}(\mathcal{H}) \vee t = B)$ 
     then break
7:    $d_t \leftarrow \text{DIAGNOSE}_{\phi}(z, \hat{y}_t, \ell_t)$ 
8:    $p_t \leftarrow e_t$ ; if  $\phi = \text{ONLINE}$  then  $p_t \leftarrow$ 
     SELECTREWRITE $_{\phi}(\mathcal{A}, \pi)$ 
9:    $s_t \leftarrow \text{OPTIMIZE}_{\phi}(s_{p_t}, d_t)$ 
10:   $f_t \leftarrow q_t$ ;  $\mathcal{A} \leftarrow \mathcal{A} \cup \{(s_t, p_t, f_t)\}$ 
11: end for
12: return FINALIZE $_{\phi}(\mathcal{A}, \mathcal{H})$ 

```

Here τ controls the sensitivity to stored search credits, c_i counts how often node i has previously been selected by the softmax policy, and β penalizes repeated selection of the same node. This policy allows STEINS to select earlier archive nodes as rewrite parents, enabling branching in the specification history while biasing selection toward higher-credit states.

3.4 Offline Corpus-Level Evolution

In the offline phase, STEINS learns a reusable corpus-level specification from a training split with reference translations and selects the final specification on a held-out validation split. At epoch t , it selects a specification from the archive, executes it on the training set, scores the outputs with the reference-based metric m , diagnoses informative failures, and rewrites the selected specification accordingly. To prioritize the most informative regressions, STEINS maintains a per-sample ratchet threshold. Let $\tilde{y}_j$ denote the original bootstrap translation and define $b_j^{(0)} = m(\tilde{y}_j, y_j)$. Let p_t denote the parent selected at epoch t , and let $\hat{y}_j^{(t)}$ be the translation obtained by executing that parent. The ratchet state inherited with the new node is updated as

$$b_j^{(t)} = \max\{b_j^{(p_t)}, m(\hat{y}_j^{(t)}, y_j)\}.$$

A sample is treated as regressing if

$$m(\hat{y}_j^{(t)}, y_j) < b_j^{(p_t)}.$$

The top- k samples with the largest regression magnitudes are selected for diagnosis. For each selected

	Offline Evolution	Online Adaptation
Scope	Training split	Single test instance
Archive scope	Corpus-shared	Instance-local
Search signal	Reference-based m	Reference-free m_{qe}
Diagnosis target	Top- k regressions	Each continuing instance
Optimizer evidence	Cross-sample aggregation	Instance report + QE status
Goal	Reusable corpus-level specification	Instance-adapted specification
Stopping rule	Fixed E -epoch budget	Convergence or R -round budget
Output rule	Validation-selected specification	Highest- m_{qe} executed translation
Reference use	Search and selection	Reporting only

Table 1: Phase-specific instantiations of STEINS.

sample, the diagnoser receives the source and reference, the ratchet-baseline and current translations with their XCOMET scores, and the process log. The resulting reports are aggregated into a structured summary for the optimizer. After optimization, a review agent abstracts example-specific edits into reusable translation strategies before the updated specification is added to the archive.

3.5 Online Instance-Level Adaptation

Online adaptation performs an independent per-input specification search. For each test input x , STEINS starts from s_0 and runs the search for at most R rounds, maintaining an input-specific local archive. Search is guided by the QE metric m_{qe} . The diagnoser receives the source text, the baseline translation, the previous-round translation, the current translation, and the process log. The optimizer receives the diagnostic evidence together with two categorical QE-derived status signals: the current quality relative to the previous round (*improved*, *degraded*, or *unchanged*) and relative to the baseline (*above*, *below*, or *equal*). Let $\hat{y}_r$ denote the translation produced by the specification executed at round r , and let $q_r = m_{\text{qe}}(\hat{y}_r)$ denote its observed round-level quality. If search continues after round r , STEINS diagnoses $\hat{y}_r$, selects an archived specification for rewriting, and appends the resulting child to the local archive with $f_{\text{new}} \leftarrow q_r$ as its search credit. The credit comes from the current round; the child is evaluated in the next round. Thus, execution proceeds linearly through successively created specifications, while rewrite selection may branch from earlier archive nodes. Early stopping is triggered once the trajectory stabilizes: an instance is considered converged at round r if $q_r \geq q_1$ and $|q_j - q_{j-1}| < \epsilon$ for every $j \in \{r - P + 1, \dots, r\}$, where P is the number of consecutive stable changes required for convergence. Let $R_x \leq R$ denote the number of executed rounds for input x . The final output is selected by

$$r^* = \arg \max_{1 \leq r \leq R_x} q_r, \quad \hat{y}^* = \hat{y}_{r^*}. \quad (1)$$

Source: In five minutes more the cloud of bewilderment dissolved: I knew quite well that I was in my own bed, and that the red glare was the nursery fire. It was night: a candle burnt on the table; Bessie stood at the bed-foot with a basin in her hand, and a gentleman sat in a chair near my pillow, leaning over me.	Evolved specification: (...) Use specific Chinese equivalents for terms like 'nursery fire' to convey the intended nuance (e.g., '育儿室的火光' or '育婴室的火光'). - Prioritize idiomatic and natural translations over literal equivalents, especially for terms like 'nursery fire'. Specifically, for terms like 'bed-foot', use precise equivalents such as '床尾' to accurately convey the intended spatial relationship, avoiding general terms like '床边' which may lead to confusion. (...)
Original specification output: 再过五分钟，那困惑的迷雾便消散了：我清楚地知道自己躺在自己的床上，那红色的光芒是 nursery 的火光。是夜晚，桌上点着一支蜡烛；贝西站在床边，手里拿着一个盆子，一位绅士坐在床边的椅子上，俯身朝我这边。	Evolved specification output: 再过五分钟，困惑的迷雾便消散了：我清楚地知道自己躺在自己的床上，那红色的光芒是育婴室的火光。是夜晚，桌上点着一支蜡烛；贝西站在床尾，手里拿着一个盆子，一位绅士则坐在我的枕头旁的椅子上，俯身朝我这边。

Figure 2: A representative online adaptation case study under CoT setting. STEINS diagnoses an input-specific translation failure in the Round 1 output and revises the specification accordingly, yielding a better final translation.

Algorithm 1 summarizes the shared search skeleton of STEINS; phase-specific instantiations for offline evolution and online adaptation are given in Appendix A.1, together with an end-to-end online search example in Figure 4. Table 1 summarizes the differences between the two phases.

4 Experiment Setup

4.1 Datasets and Language Pairs

We pair WMT23 with LITEVAL-CORPUS in a matched general-to-literary setting. WMT23 is used for both offline corpus-level evolution and online instance-level adaptation in the general-domain setting, while LITEVAL-CORPUS is used only for online instance-level adaptation in the literary setting. We restrict all experiments to the three language pairs shared by both datasets: English→Chinese (en-zh), English→German (en-de), and German→English (de-en).

WMT23 General MT. WMT23 (Kocmi et al., 2023) serves as the general-domain, segment-level benchmark in our study. Since offline evolution is multi-epoch and online adaptation performs an independent multi-round search for each test segment, we restrict the study to a controlled subset consisting of the 400 longest segments for each language pair from the WMT23 test set. We then partition this subset into two disjoint portions: 200 segments are used for offline evolution (with an 85/15 train/validation split), and the remaining 200 are held-out test segments used to evaluate both the standalone effect of offline evolution and segment-level online adaptation.

LITEVAL-CORPUS. We use LITEVAL-CORPUS (Zhang et al., 2025b) as our primary

literary benchmark and main stress test for STEINS. Literary translation places particular demands on stylistic control, discourse coherence, and culturally grounded rendering, making it a strong testbed for instance-level specification adaptation. LITEVAL-CORPUS provides verified human translations at the paragraph level, which makes it well-suited for evaluating these phenomena beyond general-domain segment translation. We evaluate the same three language pairs as in WMT23 to maintain a matched setting (The dataset statistic of LITEVAL-CORPUS is given in Appendix 7).

4.2 Translation Systems

We apply STEINS to four LLM-based translation systems spanning different agentic interaction patterns: **Single**, a single translator agent that produces the output directly from an instruction prompt; **CoT**, a single agent that reasons about the translation context before producing the final output, inspired by chain-of-thought prompting (Wei et al., 2022); **MAD**, a two-agent debate system run for three rounds, in which agents critique and revise candidate translations and a moderator selects the round output, adapted from Du et al. (2024); and **Fixed Pipeline**, a three-agent translator-reviewer-finalizer pipeline, with each agent guided by a dedicated role prompt. Full prompts and role definitions are provided in Appendix B.

4.3 Evaluation Metrics

We distinguish *optimization signals*, which drive search, from *reporting metrics*, which are used only for final evaluation. For each setting, we report not only the optimization signal itself, but also additional reference-based and reference-free metrics, allowing us to test whether gains transfer beyond

Model	Policy	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta\text{GEMBA}\uparrow$
Qwen3-8B	Sequential	$+0.33 \pm 0.44$ (11/12)	$+0.41 \pm 0.71$ (7/12)	$+0.40 \pm 0.60$ (9/12)
	Softmax	$+0.37 \pm 0.38$ (10/12)	$+0.57 \pm 0.65$ (9/12)	$+0.47 \pm 0.67$ (8/12)
GPT-4o-mini	Sequential	$+0.12 \pm 0.37$ (7/12)	-0.13 ± 0.32 (4/12)	$+0.07 \pm 0.27$ (7/12)
	Softmax	$+0.25 \pm 0.65$ (9/12)	$+0.15 \pm 0.60$ (5/12)	$+0.06 \pm 0.21$ (5/12)

Table 2: Corpus-level effects of offline-selected initialization on the WMT23 held-out test set. Values are macro-averaged differences ($\Delta = \text{Trained} - \text{Origin}$) over 12 settings (4 modes $\times$ 3 language pairs).

the signal used for refinement.

Optimization signals. Offline corpus-level evolution is performed on reference-available data, so we use XCOMET (Guerreiro et al., 2024) as the optimization signal for scoring and validation-based specification selection. Online instance-level adaptation operates directly on test inputs, with XCOMET-QE serving as the optimization signal for refinement and best-of- R selection, without reference use during search.

Reporting metrics. For offline evolution on WMT23, we report XCOMET-QE and GEMBA- DA_{ref} (Kocmi and Federmann, 2023) in addition to the optimization signal XCOMET. For online adaptation on WMT23, where XCOMET-QE is the optimization signal, we additionally report XCOMET and GEMBA- DA_{ref} . For online adaptation on LITEVAL-CORPUS, where XCOMET-QE is the optimization signal, we additionally report XCOMET and LiTransProQA (V_w) (Zhang et al., 2025c), a literary-focused reference-free QA evaluator targeting discourse and stylistic dimensions such as tone, figurative language, and cultural rendering, with dimension weights informed by professional literary translators. Additional evaluator details are provided in Appendix F.

4.4 Implementation Details

We use two backbone LLMs for all settings: **Qwen3-8B** (Qwen Team, 2025), served locally via vLLM (Kwon et al., 2023), and **GPT-4o-mini** (OpenAI, 2024), accessed through the OpenAI API. Within each run, the same backbone is used for translation, diagnosis, and optimization, and all generation temperatures are set to zero to reduce sampling variance. Across both phases, we compare two rewrite-selection policies: **Sequential** and **Softmax**, with temperature $\tau=0.1$ and novelty penalty $\beta=0.5$ for Softmax sampling. The temperature sensitivity is further explored in Appendix G.

Offline corpus-level evolution runs for $E=20$

epochs. At each epoch, we diagnose the top- k samples with the largest regressions ($k=32$). For online instance-level adaptation, each input undergoes up to $R=20$ refinement rounds. Early stopping is triggered once the online optimization signal is at least as high as the Round 1 baseline and changes by less than $\epsilon=0.01$ for $P=3$ consecutive rounds. We evaluate two initializations: **Trained** (offline-evolved) specification and **Origin** (hand-crafted) specification. All other metrics are computed on the output selected by XCOMET-QE for each input, rather than on metric-specific optimal rounds, ensuring that reported results reflect the actual system output rather than an oracle upper bound. Main-text results report macro-averages across 12 settings (4 architectures $\times$ 3 language pairs), with full per-setting results in Appendix I.

5 Results and Analysis

We systematically study offline corpus-level evolution, online instance-level adaptation, and their combination through different starting specifications. Across both benchmarks, two findings stand out: offline evolution yields modest but generally positive gains, whereas online adaptation delivers much larger and more consistent improvements across datasets, backbones, and evaluators. The following analysis examines the standalone value of offline evolution, its role as a starting point for online adaptation, and the quality–efficiency trade-offs induced by initialization and rewrite selection.

5.1 Offline Corpus-Level Evolution

Table 2 shows that offline corpus-level evolution yields modest but generally positive gains on held-out WMT23 test split. Improvements are larger and more consistent for Qwen3-8B, whereas gains for GPT-4o-mini are smaller and more mixed, especially on XCOMET-QE. Overall, the results suggest that offline evolution learns a transferable corpus-level improvement, but one whose effect remains limited and backbone-sensitive. This pattern

WMT23 (segment-level)					
Init	Model	Policy	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta\text{GEMBA}\uparrow$
Origin	Qwen3-8B	Sequential	$+2.51 \pm 1.06$ (12/12)	$+3.84 \pm 1.88$ (12/12)	$+0.72 \pm 0.72$ (11/12)
		Softmax	$+2.78 \pm 1.30$ (12/12)	$+4.36 \pm 2.19$ (12/12)	$+0.78 \pm 0.73$ (11/12)
	GPT-4o-mini	Sequential	$+1.41 \pm 0.71$ (12/12)	$+2.45 \pm 1.34$ (12/12)	$+0.19 \pm 0.17$ (9/12)
		Softmax	$+1.66 \pm 0.87$ (12/12)	$+2.77 \pm 1.58$ (12/12)	$+0.21 \pm 0.21$ (11/12)
Trained	Qwen3-8B	Sequential	$+2.10 \pm 1.05$ (12/12)	$+3.24 \pm 1.69$ (12/12)	$+0.49 \pm 0.64$ (9/12)
		Softmax	$+2.25 \pm 1.23$ (12/12)	$+3.69 \pm 2.05$ (12/12)	$+0.56 \pm 0.54$ (12/12)
	GPT-4o-mini	Sequential	$+1.28 \pm 0.83$ (12/12)	$+2.35 \pm 1.40$ (12/12)	$+0.08 \pm 0.16$ (7/12)
		Softmax	$+1.32 \pm 0.92$ (12/12)	$+2.43 \pm 1.48$ (12/12)	$+0.12 \pm 0.17$ (9/12)
LITEVAL-CORPUS (paragraph-level)					
Init	Model	Policy	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta V_w \uparrow$
Origin	Qwen3-8B	Sequential	$+2.77 \pm 1.37$ (11/12)	$+4.71 \pm 0.87$ (12/12)	$+0.06 \pm 0.07$ (9/12)
		Softmax	$+3.49 \pm 1.25$ (12/12)	$+5.31 \pm 1.02$ (12/12)	$+0.09 \pm 0.08$ (11/12)
	GPT-4o-mini	Sequential	$+2.60 \pm 1.13$ (12/12)	$+3.96 \pm 0.92$ (12/12)	$+0.04 \pm 0.03$ (11/12)
		Softmax	$+2.57 \pm 0.91$ (12/12)	$+3.83 \pm 1.09$ (12/12)	$+0.03 \pm 0.04$ (9/12)
Trained	Qwen3-8B	Sequential	$+2.50 \pm 1.05$ (12/12)	$+4.34 \pm 1.20$ (12/12)	$+0.05 \pm 0.06$ (10/12)
		Softmax	$+2.93 \pm 0.99$ (12/12)	$+4.72 \pm 0.88$ (12/12)	$+0.07 \pm 0.06$ (10/12)
	GPT-4o-mini	Sequential	$+2.25 \pm 0.64$ (12/12)	$+3.74 \pm 0.99$ (12/12)	$+0.01 \pm 0.03$ (8/12)
		Softmax	$+2.02 \pm 0.96$ (12/12)	$+3.49 \pm 0.95$ (12/12)	$+0.01 \pm 0.04$ (7/12)

Table 3: Macro-averaged gains from online instance-level adaptation on WMT23 and LITEVAL-CORPUS. Values report improvements of the final output over the Round-1 baseline across 12 settings, shown as mean $\pm$ std; all metrics are computed on the same output, and parentheses indicate the number of settings with positive gains.

is consistent with the difficulty of compressing heterogeneous translation failures into a single shared corpus-level update.

5.2 Online Instance-Level Adaptation

Table 3 shows that online instance-level adaptation yields large and highly consistent gains under its QE-based optimization signal. On WMT23, XCOMET-QE improves by +2.35 to +4.36, while XCOMET improves by +1.28 to +2.78; both metrics are positive in 12/12 settings for every condition. These gains transfer to paragraph-level literary translation: on LITEVAL-CORPUS, XCOMET-QE improves by +3.49 to +5.31 and XCOMET by +2.02 to +3.49, again almost always positive (11/12 or 12/12 settings in most conditions). The magnitude and consistency of these improvements support our central claim that instance-level diagnose-optimize updates are well matched to translation errors that vary substantially across inputs.

The improvements also transfer beyond the QE optimization signal. On WMT23, GEMBA improves by +0.08 to +0.78 across conditions. On LITEVAL-CORPUS, the literary-focused evalua-

tor V_w (LiTransProQA; §4.3) improves by +0.01 to +0.09, with most conditions positive. Since V_w aggregates weighted QA judgments over literary dimensions using weights informed by professional translator input, improvements on this metric provide additional evidence that the gains are not confined to optimization signal alone. Paired significance tests are reported in Appendix 18, and full per-setting results are provided in Appendix I.

5.3 Initialization Effects: Trained vs. Origin

Table 3 shows that online adaptation is effective under either initialization: starting from both **Origin** and **Trained**, gains are consistently positive across datasets and backbones. At the same time, **Origin** often yields larger improvements, indicating greater adaptation headroom when starting from the unevolved specification.

Table 4 shows that **Trained** is typically more efficient and more stable. In most settings, it requires fewer refinement rounds and achieves higher convergence rates than **Origin**. Taken together, these results clarify the role of offline corpus-level evolution in STEINS: it is useful both as a modest standalone corpus-level improvement and as an effi-

Data	Model	Pol.	Avg Rounds		Conv %	
			O	$\Delta\downarrow$	O	$\Delta\uparrow$
WMT23	Qwen3-8B	Seq	13.0	-1.4	57.9	+8.0
	Qwen3-8B	Soft	14.3	-1.5	44.3	+9.0
	GPT-4o-mini	Seq	11.0	-0.5	69.5	+3.2
	GPT-4o-mini	Soft	11.7	-0.4	62.1	+3.0
LitEval	Qwen3-8B	Seq	15.6	-1.0	43.5	+7.7
	Qwen3-8B	Soft	16.9	-0.6	28.0	+2.0
	GPT-4o-mini	Seq	13.5	-0.3	55.1	+3.1
	GPT-4o-mini	Soft	14.2	+0.1	47.9	-1.1

Table 4: Efficiency and convergence of online adaptation. “O” denotes the macro-averaged value under Origin initialization, and Δ denotes Trained – Origin over the same 12 settings.

cient initialization for subsequent online adaptation. Its main benefit in the combined setting is therefore not larger adaptation headroom, but a better starting point that often reaches strong performance with less test-time computation. At deployment scale, even modest per-instance savings can accumulate into meaningful reductions in total test-time computation.

5.4 Search Policy and Efficiency Trade-offs

Softmax sampling can improve online adaptation quality, but its gains are strongly conditioned by initialization and backbone. In Table 3, Softmax is generally more favorable under **Origin**, where adaptation headroom is larger; this tendency is clearest on WMT23 for both backbones and on LitEval for Qwen3-8B. Under **Trained**, or for GPT-4o-mini on LITEVAL-CORPUS, the gains become smaller or disappear. Full pairwise policy deltas are reported in Appendix Table 19.

Table 4 shows that Softmax typically requires more rounds and converges less reliably than Sequential. This is consistent with its broader exploration behavior: by revisiting strong historical nodes rather than following only the most recent trajectory, it can sometimes recover better solutions from earlier branches. The trade-off is most favorable when the starting specification is weak and substantial headroom remains, but less favorable once initialization is already strong. In practice, **Trained+Sequential** is the default choice for compute-limited deployment, whereas **Origin+Softmax** is preferable when maximizing translation quality justifies additional test-time search.

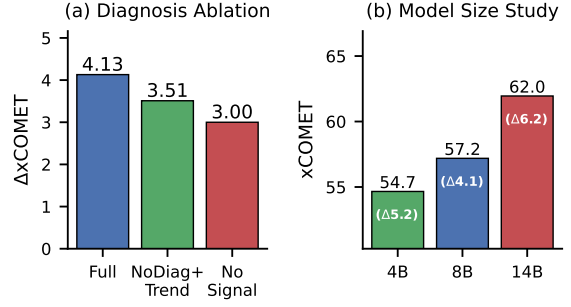


Figure 3: Ablation and scaling analyses of STEINS: (a) update-signal ablation; (b) model scaling.

5.5 Qualitative Example

Figure 2 illustrates a representative case of online adaptation. The initial translation contains two local problems that are especially harmful in literary narration: an under-localized rendering of a scene-setting term, which breaks target-language naturalness, and an imprecise description of character placement, which weakens the spatial clarity of the scene. The evolved specification addresses both issues by promoting a more idiomatic Chinese rendering and tighter spatial grounding, producing a translation that reads as more natural, precise, and narratively coherent.

5.6 Ablation and Scaling Analyses

Figure 3 reports two targeted analyses on LITEVAL-CORPUS en-zh under online adaptation with **Origin** initialization, using the CoT system. Figure 3(a) compares $\Delta XCOMET$ under three update variants: **Full**, the complete STEINS setup; **NoDiag+Trend**, which removes the MQM-style diagnostic report but retains round-to-round trend signals; and **No Signal**, which allows unconstrained prompt rewriting without either form of feedback. The ordering **Full** > **NoDiag+Trend** > **No Signal** indicates that gains come not merely from iterative rewriting, but from injecting translation-specific, span-grounded diagnostic evidence into specification updates. Figure 3(b) examines model scaling under the same setting. Final XCOMET increases monotonically from Qwen3-4B to 8B to 14B (54.7, 57.2, and 62.0, respectively), showing that STEINS remains complementary to stronger backbones. The 4B model still benefits substantially from online adaptation, indicating that STEINS remains effective even for weaker starting outputs. Full results for all metrics are reported in Appendix E.

Method	Trans.	Eval/QE	Diag.+Opt.	Total
zero-shot	n	0	0	n
offline	nE	nE	$(k+2)E$	$(2n+k+2)E$
online	nR	nR	$2nR$	$4nR$

Table 5: LLM-call complexity. n : number of sentences; E : offline training epochs; R : average online adaptation rounds per sentence; k : diagnosed samples per epoch.

5.7 Inference Cost and Adaptation Efficiency

The iterative nature of self-evolving agentic systems makes computational efficiency an important practical consideration. Table 5 summarizes the inference complexity of fixed prompting, offline evolution, and online adaptation, while Table 6 reports measured per-segment online costs in terms of adaptation rounds, LLM calls, wall-clock time, and token usage. Online QE evaluations are batched across active segments within each round; thus, per-segment call counts reflect total inference work rather than sequential wall-clock execution.

The cost varies substantially across agent architectures. MAD is the most expensive setting, averaging 140.9 LLM calls and 270.8 seconds per segment, reflecting its multi-agent debate structure and retained conversational context. Nevertheless, the gain trajectories in Appendix Table 13 show that the first five rounds capture approximately 75% of the total quality improvement across all four modes, suggesting that a smaller adaptation budget can preserve much of the gain while substantially reducing inference cost.

6 Conclusion

We introduced STEINS, a self-evolving framework for agentic machine translation that optimizes agent specifications through a diagnose-optimize loop using MQM-style, span-grounded error evidence. This design enables targeted and interpretable specification updates without model parameter tuning. Across WMT23 and LITEVAL-CORPUS, offline corpus-level evolution provides modest but generally positive gains and a useful warm start, whereas online instance-level adaptation delivers the largest and most consistent improvements, highlighting the value of instance-level specification adaptation in agentic machine translation.

Limitations

Our framework introduces a quality-efficiency trade-off at inference time: online instance-level

Mode	Avg R	Calls	Wall (s)	In tok.	Out tok.
Single	13.85	39.3	13.5	53,201	11,882
CoT	13.76	39.3	32.0	49,110	14,201
Fixed	14.04	69.2	68.3	71,513	17,034
MAD	13.29	140.9	270.8	147,434	33,295

Table 6: Per-segment measurements for an online run (Qwen3-8B, en-zh, WMT23). Calls and wall-clock include LLM inference only.

adaptation requires multiple rounds of translation, evaluation, and diagnosis, increasing latency and cost despite early stopping. Online updates also depend on a reference-free quality estimator (XCOMET-QE), which may introduce metric-specific biases; although gains generally transfer to independent evaluators, improvements are not guaranteed for every quality dimension. Offline corpus-level evolution yields only modest gains in our experiments and remains backbone-sensitive, suggesting that a single global specification is useful but insufficient for fully addressing heterogeneous, input-specific translation errors. In addition, diagnosis relies on an LLM and may suffer from imperfect or inconsistent error attribution, even with span grounding and protected prompt regions. Finally, our evaluation covers only three language pairs and two benchmarks; broader validation across more language pairs, longer documents, real-world domains, and larger-scale professional human evaluation remains important future work. Future work may explore adaptive temperature schedules and more efficient search strategies to further improve the quality-efficiency trade-off.

Acknowledgments

This work was supported by JST SPRING, Grant Number JPMJSP2108 and by the "R&D Hub Aimed at Ensuring Transparency and Reliability of Generative AI Models" project of the Ministry of Education, Culture, Sports, Science and Technology. We thank all those who provided valuable feedback that helped improve this work.

References

- Meiqi Chen, Fandong Meng, Yingxue Zhang, Yan Zhang, and Jie Zhou. 2024. Crat: A multi-agent framework for causality-enhanced reflective and retrieval-augmented translation with large language models. *arXiv preprint arXiv:2410.21067*.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenen-

- baum, and Igor Mordatch. 2024. Improving factuality and reasoning in language models through multi-agent debate. In *Forty-first international conference on machine learning*.
- Himanshu Dutta, Sunny Manchanda, Prakhar Bapat, Meva Ram Gurjar, and Pushpak Bhattacharyya. 2025. Graft: A graph-based flow-aware agentic framework for document-level machine translation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 2405–2428.
- Zhaopeng Feng, Shaosheng Cao, Jiahao Ren, Jiayuan Su, Ruizhe Chen, Yan Zhang, Zhe Xu, Yao Hu, Jian Wu, and Zuozhu Liu. 2025. Mt-r1-zero: Advancing llm-based machine translation via r1-zero-like reinforcement learning. *arXiv preprint arXiv:2504.10160*.
- Zhaopeng Feng, Yan Zhang, Hao Li, Bei Wu, Jiayu Liao, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2024. Tear: Improving llm-based machine translation with systematic self-refinement. *arXiv preprint arXiv:2402.16379*.
- Chrisantha Fernando, Dylan Banarse, Henryk Michalewski, Simon Osindero, and Tim Rocktäschel. 2023. Promptbreeder: Self-referential self-improvement via prompt evolution. *arXiv preprint arXiv:2309.16797*.
- 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.
- Irving John Good. 1966. Speculations concerning the first ultraintelligent machine. In *Advances in computers*, volume 6, pages 31–88. Elsevier.
- Nuno M Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André FT Martins. 2024. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.
- Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. 2023. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. *arXiv preprint arXiv:2309.08532*.
- Mingui He, Yilun Liu, Shimin Tao, Yuanchang Luo, Hongyong Zeng, Chang Su, Li Zhang, Hongxia Ma, Daimeng Wei, Weibin Meng, and 1 others. 2025. R1-t1: Fully incentivizing translation capability in llms via reasoning learning. *arXiv preprint arXiv:2502.19735*.
- Shengran Hu, Cong Lu, and Jeff Clune. 2024. Automated design of agentic systems. *arXiv preprint arXiv:2408.08435*.
- Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Masaaki Nagata, Toshiaki Nakazawa, Martin Popel, and 3 others. 2023. [Findings of the 2023 conference on machine translation \(WMT23\): LLMs are here but not quite there yet](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 1–42, Singapore. Association for Computational Linguistics.
- Tom Kocmi and Christian Federmann. 2023. [Large language models are state-of-the-art evaluators of translation quality](#). In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 193–203, Tampere, Finland. European Association for Machine Translation.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pages 611–626.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *arXiv preprint arXiv:2305.19118*.
- Alexander Novikov, Ngan Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco JR Ruiz, Abbas Mehrabian, and 1 others. 2025. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*.
- OpenAI. 2024. [GPT-4o mini: advancing cost-efficient intelligence](#). Accessed: 2026-01-14.
- Qwen Team. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Jürgen Schmidhuber. 1987. *Evolutionary principles in self-referential learning, or on learning how to learn: the meta-meta-... hook*. Ph.D. thesis, Technische Universität München.
- Jürgen Schmidhuber. 2007. Gödel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence*, pages 199–226. Springer.
- Jiaan Wang, Fandong Meng, Yunlong Liang, and Jie Zhou. 2024a. Drt: Deep reasoning translation via long chain-of-thought. *arXiv preprint arXiv:2412.17498*.
- Yutong Wang, Jiali Zeng, Xuebo Liu, Derek F Wong, Fandong Meng, Jie Zhou, and Min Zhang. 2024b. Delta: An online document-level translation agent based on multi-level memory. *arXiv preprint arXiv:2410.08143*.

- 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.
- Minghao Wu, Jiahao Xu, Yulin Yuan, Gholamreza Haffari, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. (perhaps) beyond human translation: Harnessing multi-agent collaboration for translating ultra-long literary texts. *arXiv preprint arXiv:2405.11804*.
- Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2023. Large language models as optimizers. In *The Twelfth International Conference on Learning Representations*.
- Xunjian Yin, Xinyi Wang, Liangming Pan, Li Lin, Xiaojun Wan, and William Yang Wang. 2024. G\ "odel agent: A self-referential agent framework for recursive self-improvement. *arXiv preprint arXiv:2410.04444*.
- Mert Yuksekogonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic "differentiation" via text. *arXiv preprint arXiv:2406.07496*.
- Jenny Zhang, Shengran Hu, Cong Lu, Robert Lange, and Jeff Clune. 2025a. Darwin godel machine: Open-ended evolution of self-improving agents. *arXiv preprint arXiv:2505.22954*.
- Ran Zhang, Wei Zhao, and Steffen Eger. 2025b. [How good are LLMs for literary translation, really? literary translation evaluation with humans and LLMs](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10961–10988, Albuquerque, New Mexico. Association for Computational Linguistics.
- Ran Zhang, Wei Zhao, Lieve Macken, and Steffen Eger. 2025c. Litransproqa: an llm-based literary translation evaluation metric with professional question answering. *arXiv preprint arXiv:2505.05423*.
- Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. 2022. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*.

A Appendix

A.1 Implementation of the Offline and Online Phase

Algorithm 2 Offline Corpus-Level Evolution

Require: training corpus $\mathcal{D}_{\text{tr}} = \{(x_i, y_i)\}_{i=1}^n$, validation corpus $\mathcal{D}_{\text{val}}$, initial specification s_0 , rewrite-selection policy π , epochs E , diagnostic budget k

Ensure: selected specification s^*

```

1:  $\mathcal{A} \leftarrow \{(s_0, \perp, 0)\}$ ;  $\mathcal{B}_0 \leftarrow \text{BOOTSTRAP}(\mathcal{D}_{\text{tr}})$ 
2: for  $e = 1$  to  $E$  do
3:    $u_e \leftarrow \text{SELECTPARENT}(\mathcal{A}, \pi)$ 
4:   for  $i = 1$  to  $n$  do
5:      $(\hat{y}_i^{(e)}, \ell_i^{(e)}) \leftarrow$ 
        $\text{TRANSLATEWITHLOG}(x_i, s_{u_e}, \tilde{y}_i^{(u_e)})$ 
6:      $q_i^{(e)} \leftarrow \text{XCOMET}(x_i, \hat{y}_i^{(e)}, y_i)$ 
7:      $\delta_i^{(e)} \leftarrow b_i^{(u_e)} - q_i^{(e)}$ 
8:   end for
9:    $\mathcal{I}_e \leftarrow \text{TOPK}(\{i : \delta_i^{(e)} > 0\}, \delta^{(e)}, k)$ 
10:   $\mathcal{R}_e \leftarrow \text{DIAGNOSEMQM}(\mathcal{I}_e, \mathcal{D}_{\text{tr}}, \mathcal{B}_{u_e}, \hat{\mathbf{y}}^{(e)}, \mathbf{q}^{(e)}, \ell^{(e)})$ 
11:   $\Gamma_e \leftarrow \text{AGGREGATEDIAGNOSTICS}(\mathcal{R}_e)$ 
12:   $s_e \leftarrow \text{REVIEW}(\text{UPDATESPEC}(s_{u_e}, \Gamma_e))$ 
13:   $f_e \leftarrow \frac{1}{n} \sum_{i=1}^n q_i^{(e)}$ 
14:   $\mathcal{A} \leftarrow \mathcal{A} \cup \{(s_e, u_e, f_e)\}$ 
15:   $\mathcal{B}_e \leftarrow \text{RATCHETUPDATE}(\mathcal{B}_{u_e}, \hat{\mathbf{y}}^{(e)}, \mathbf{q}^{(e)})$ 
16: end for
17:  $e^* \leftarrow \arg \max_{e \in [E]}$ 
    $\text{EVALUATEONVALIDATION}(s_{u_{e^*}}, \mathcal{D}_{\text{val}})$ 
18: return  $s_{u_{e^*}}$ 

```

Note. $\mathcal{B}_u = \{(\tilde{y}_i^{(u)}, b_i^{(u)})\}_{i=1}^n$ is the branch-local ratchet state, initialized from the fixed zero-shot baseline. The execution of s_{u_e} supplies the search credit f_e stored with child s_e ; it is not an independent evaluation of s_e . Validation is applied post hoc to specifications executed during evolution.

Algorithm 3 Online Instance-Level Adaptation

Require: input instance x , starting specification s_0 , rewrite-selection policy π , max rounds R , stability threshold ϵ , patience P

Ensure: best-of- R translation $\hat{y}^*$

```

1:  $\mathcal{A}_x \leftarrow \{(s_0, \perp, 0)\}$ ;  $s_{\text{cur}} \leftarrow s_0$ ;  $\hat{y}^{(0)} \leftarrow \emptyset$ 
2:  $\hat{y}^* \leftarrow \emptyset$ ;  $q^* \leftarrow -\infty$ ;  $c_{\text{stable}} \leftarrow 0$ 
3: for  $r = 1$  to  $R$  do
4:    $(\hat{y}^{(r)}, \ell^{(r)}) \leftarrow \text{TranslateWithLog}(x, s_{\text{cur}}, \hat{y}^{(r-1)})$ 
5:    $q_r \leftarrow \text{XCOMET-QE}(x, \hat{y}^{(r)})$ 
6:   if  $r = 1$  then  $q_0 \leftarrow q_1$ 
7:   if  $q_r > q^*$  then
8:      $(\hat{y}^*, q^*) \leftarrow (\hat{y}^{(r)}, q_r)$ 
9:   end if
10:  if  $r > 1$  then
11:     $c_{\text{stable}} \leftarrow c_{\text{stable}} + 1$  if  $|q_r - q_{r-1}| < \epsilon$ ; otherwise  $c_{\text{stable}} \leftarrow 0$ 
12:  end if
13:  if  $r = R$  or  $(r > 1 \wedge q_r \geq q_1 \wedge c_{\text{stable}} \geq P)$  then break
14:     $\mathcal{R}^{(r)} \leftarrow \text{DiagnoseMQM}(x, \hat{y}^{(1)}, \hat{y}^{(r-1)}, \hat{y}^{(r)}, \ell^{(r)})$ 
15:     $(z_{\text{prev}}^{(r)}, z_{\text{base}}^{(r)}) \leftarrow \text{ScoreStatus}(q_r, q_{r-1}, q_1)$ 
16:     $u_r \leftarrow \text{SelectRewriteParent}(\mathcal{C}_r, \pi)$ 
17:     $s_{\text{new}} \leftarrow \text{UpdateSpec}(s_{u_r}, \mathcal{R}^{(r)}, z_{\text{prev}}^{(r)}, z_{\text{base}}^{(r)})$ 
18:     $\mathcal{A}_x \leftarrow \mathcal{A}_x \cup \{(s_{\text{new}}, u_r, q_r)\}$ ;  $s_{\text{cur}} \leftarrow s_{\text{new}}$ 
19:  end for
20: return  $\hat{y}^*$ 

```

Note. The diagnoser observes the baseline, previous, and current translations and the process log, but no QE scores; the optimizer receives its report together with categorical QE status signals. For sequential selection, the rewrite parent is s_{cur} . For softmax selection, $\mathcal{C}_r$ contains the children generated before the round- r update; sampling is used only when $|\mathcal{C}_r| \geq 2$, and otherwise defaults to s_{cur} . The execution at round r supplies the search credit q_r stored with the resulting child; it is not an independent evaluation of that child, which is executed in the following round.

B System Prompt Templates

We list the original (hand-crafted) agent specifications for each translation architecture below. Placeholders are shown in **{violet}**. Shaded regions denote protected zones that the optimizer cannot modify (§3.2).

B.1 Zero-Shot Translation Prompt

Zero-Shot Prompt

Translate the following text from **{src_lng}** to **{tgt_lng}**: **{source}**. Please return only the **{tgt_lng}** translation. Do not include JSON, metadata, explanations, or any additional content.

B.2 Single-Agent

Agent Prompt

You are a translation expert.

Translate the following text from **{src_lng}** to **{tgt_lng}**: **{source}**. Please return only the **{tgt_lng}** translation.

Do not include JSON, metadata, explanations, or any additional content.

B.3 Chain-of-Thought (CoT)

CoT Prompt

You are a translation expert.

Translate the following text from **{src_lng}** to **{tgt_lng}**: **"{source}"**. Strictly return JSON: **{"thinking": "...", "translation": "..."}"**

You are a careful human translation expert: you first form a full understanding of the sentence, then produce a clean final translation.

Before writing the final translation, briefly reason internally about:

- What the sentence is doing (statement / instruction / warning / contrast / definition).
 - Key entities, numbers, and technical terms that must stay exact.
 - Ambiguous parts (pronouns, tense / aspect, scope of negation, modifiers) and resolve them using context if provided.
 - Style and register (formal / informal, academic/ marketing) and keep it consistent.
- Then write the final translation in one pass, fluent and faithful, with no extra text.

B.4 Fixed Pipeline

Translator Prompt

You are a professional translator. Focus on accuracy and natural expression.

Translate the following text accurately and naturally: Source (**{src_lng}**): **{source}**
Provide only the **{tgt_lng}** translation.

Just provide the translation, do not include explanations.

Reviewer Prompt

You are a translation reviewer. Your job is to improve translations for accuracy, fluency, and cultural appropriateness.

Review and improve the following draft translation: Source (**{src_lng}**): **{source}**
Draft translation: **{draft_translation}**
Provide an improved **{tgt_lng}** translation.

Just provide the translation, do not include explanations.

Finalizer Prompt

You are the final quality controller in a translation pipeline. Your task is to produce the best possible translation.

Compare the translations and produce the final polished version:
Source (**{src_lng}**): **{source}**
Draft: **{draft_translation}**
Reviewed: **{reviewed_translation}**
Provide the final **{tgt_lng}** translation.

Just provide the translation, do not include explanations.

B.5 Multi-Agent Debate (MAD)

Player Meta Prompt (Debater Identity)

You are a debater participating in a translation debate competition. Welcome to this constructive and thoughtful debate. It is entirely appropriate to offer different perspectives, as our shared objective is to discover the most accurate and natural translation.

The debate topic is as follows: What is the appropriate **{tgt_lng}** translation of the following **{src_lng}** text: **"{source}"** **{context}**.

Moderator Meta Prompt (Moderator Identity)

You are acting as a moderator of a translation debate competition. In this translation debate, debaters will present their proposed translations and explain their reasoning.

Regarding the correct `{tgt_lng}` version of the given `{src_lng}` text: `{source}`. `{context}`

At the conclusion of each round, your role is to synthesize and optimize the translations.

Initial Prompt (First Round)

Please translate the following sentence to `{tgt_lng}`: `{source}`
Afterwards, please briefly explain your reasoning.
Strictly return JSON: `{"Translation": "", "Brief Reason": ""}`
Only return the JSON object.

Debate Prompt (Subsequent Rounds)

Critically evaluate other agents' translations and refine your own to produce the most accurate and natural result.

Please translate the following sentence to `{tgt_lng}`: `{source}`
If it is helpful, here are some suggested translations from other agents:
`{candidate_solutions}`
`{pre_mod}`
Strictly return JSON: `{"Translation": "", "Brief Reason": ""}`

Only return the JSON object. If an agent's reasoning is incorrect, point it out. State the answer FIRST, then your reasoning.

Moderator Prompt

As the moderator, evaluate all candidate translations and select or synthesize the best one based on accuracy, fluency, and naturalness.

The `{round}` round of the translation debate has concluded.
If it is helpful, here are some suggested translations from other agents:
`{candidate_solutions}`
Strictly return JSON: `{"Translation": "", "Brief Reason": ""}`

You can select the best translation from the candidates, or refine one to improve it. You may also reject all candidates and provide your own. But you must return only one final translation.
State the answer FIRST, then your reasoning. Only return the JSON object.

C Steins Prompts

The diagnostic module produces MQM-structured error reports. Offline and online variants differ in available evidence: the offline diagnoser receives reference translations and quality scores, while the online diagnoser operates without either.

C.1 Offline Diagnostic Prompt

Used in the offline training phase to analyze why a system translation scored lower than the baseline.

Offline Diagnostic Prompt (diagnose_sample)

You are an expert in machine translation quality diagnosis and agentic system analyzing. Your job is to explain why the system output translation scored LOWER than the baseline translation, and produce an MQM-style structured diagnostic that can be directly used by an optimizer agent to revise prompts/roles/execution plan.

INPUT

- translation framework: `{process_type}`
- Source: `{source}`
- Reference: `{reference}`
- Baseline Translation: `{base_translation}`
- System Updated Translation: `{updated_translation}`
- xCOMET Scores: `{xcomet_scores}`
- Score Pattern: `{score_pattern}`

-- Translation Process Log --
`{translation_process}`

MQM taxonomy definitions:

Accuracy, Fluency, Terminology, Style, Locale convention, Other – omitted for brevity

TASK

Step 1) Compare updated translation vs. baseline based on source vs reference. Identify why the updated translation scored worse.

Step 2) Output MQM issues with severity.

Step 3) Provide framework-aware, actionable suggestions specifically for improving this translation framework.

Severity rules:

- Major: actual errors that degrade meaning or correctness.
- Minor: smaller imperfections that preserve meaning.
- Neutral: purely subjective preference.

Reporting constraints:

- Report AT MOST 5 issues. Focus on the most important issues that likely caused the score drop.
- Suggestions must target: `system_prompt`, `agent_roles`, `execution_plan` ...
- Do NOT give generic advice. Tie each suggestion to specific MQM issues.

C.2 Online Diagnostic Prompt

Used during test-time scaling. Operates without reference translations or quality scores — diagnosis is purely evidence-based.

Online Diagnostic Prompt (diagnose_tts)

You are an expert in machine translation quality diagnosis. You do NOT have access to a reference translation or quality scores.

INPUT

```
- Source: {source}
- Baseline Translation: {base_translation}
- Previous round Translation:
  {prev_translation}
- Current round Translation:
  {curr_translation}
- Translation Framework: {process_type}

-- Translation Process Log --
{translation_process}
```

TASK

Identify issues in the current translation based on SOURCE-TARGET ALIGNMENT. Focus on EVIDENCE-BASED issues that can be verified from source text. Compare current vs previous round and baseline translations to identify:

- 1) What changed between previous and current translation? (improvements or regressions)
- 2) Any issues that persist from baseline?
- 3) Any new issues introduced?

DO NOT:

- Make subjective preference judgments without evidence
- Report issues without source evidence

[MQM taxonomy...]

C.3 Optimizer Prompts

The optimizer receives diagnostic reports and generates JSON patches to modify the editable (non-fixed) regions of translation prompts. Both variants share framework-aware constraints and the patch format; they differ in optimization objective.

C.4 Offline Optimizer Prompt

Optimizes for dataset-level generalization. Explicitly prohibits case-specific rules.

Offline Optimizer Prompt

You are an expert prompt-optimizer for agentic machine translation systems.

```
FRAMEWORK: {framework}
{framework_description}

ALLOWED_PATHS: {editable_paths}
CURRENT_CONFIG: {current_prompts}
DIAGNOSIS_SUMMARY: {aggregated_diagnosis}
```

GENERALIZATION (TRAIN MODE)

You are optimizing prompt for a dataset-level translation system, not for individual sentences. The prompt will be applied to UNSEEN data, so it MUST generalize.

CRITICAL RULES:

- NEVER include specific translations, names, or term mappings from examples.
- ALWAYS abstract diagnosed issues into general translation strategies.
- Focus on CATEGORIES of problems, not specific instances.
- NEVER build glossary-style 'source→target' term lists in ANY field.

Examples of BAD vs GOOD rules:

- BAD: "Use '[Chinese rendering]' for Bessie"

- GOOD: "Maintain consistent rendering of proper nouns throughout the text"
 - BAD: "translate 'lay to' as '[specific Chinese nautical term]'"
 - GOOD: "Use precise domain-specific terminology for nautical/technical terms"
- [Output Schema]

C.5 Online Optimizer Prompt

Optimizes for a single translation instance. Encourages case-specific rules.

Online Optimizer Prompt

You are an expert prompt-optimizer for agentic machine translation systems.

```
FRAMEWORK: {framework}
{framework_description}

ALLOWED_PATHS: {editable_paths}
CURRENT_CONFIG: {current_prompts}
DIAGNOSIS {diagnosis}

TTS OPTIMIZATION CONTEXT
STATUS: {status}
(vs. previous round and baseline)
Strategy: {strategy}
```

Strategy Matrix:

vs. Pre	vs. Base	Strategy
improved	above	REINFORCE: ...
improved	below	CONTINUE: ...
improved	equal	FINE-TUNE: ...
degraded	...	...
...	...	...
unchanged	...	...
...	...	...

SPECIFICITY

You are optimizing for a SINGLE instance. This prompt will ONLY be used for this specific translation case.

- You CAN include case-specific translation guidance.
 - Preserve terminology choices identified as good translations.
 - Be specific about style, tone, and register for this particular content.
- [Output Schema]

C.6 Review Prompt

In offline phase, a review agent post-processes the optimizer output to abstract away any residual case-specific artifacts before the prompt is applied to the next epoch.

Review Agent Prompt

You are a prompt quality reviewer for a machine translation optimization system.

BACKGROUND:

An upstream optimizer derives translation prompt rules from error diagnostics. These diagnostics inevitably include concrete instances (source segments, reference translations, annotated error spans). As

a result, the optimizer may propagate surface-level artifacts into rules.

ALLOWED_PATHS: {`editable_paths`}
CURRENT_CONFIG: {`optimizer_output`}

YOUR TASK:

- ABSTRACT:** Replace any rule that references quoted names, terms, or translations with a general pattern.
 - Before: “Use ‘Vorladung’ for ‘subpoena’ and ‘Einziehung’ for ‘seizure’”
 - After: “Use precise domain-appropriate legal terminology”
- Before: “Avoid mistranslations of phrases like ‘Amid Dropping Income’”
- After: “Translate idiomatic phrases naturally rather than literally”
- MERGE:** Combine similar or overlapping rules into a single, stronger rule.
- DEDUPLICATE:** Remove exact or near-duplicate rules.
- SIMPLIFY:** Make each rule concise.
- IMPROVE:** Rewrite vague rules to be more actionable and precise.
 - Before: “Ensure correct use of terms”
 - After: “Choose register-appropriate vocabulary: formal contexts require formal equivalents”

HARD CONSTRAINTS:

- Every rule in the output must be applicable to ANY translation text.
- Ensure zero leakage of reference translations: remove any concrete translation details, quoted terms, term mappings, or target-language text originating from reference data.
- Preserve all placeholders (##source##, ##src_lng##, etc.).

[Output Schema]

D LITEVAL-CORPUS Dataset Statistic

Pair	Paragraphs	Segments	Sentences	Sentences / Paragraph	
				Source	Target
de-en	46	562	4310	7.1	7.7
en-de	46	554	2790	4.4	5.0
de-zh	47	520	3039	5.3	5.8
en-zh	49	552	3162	5.5	5.7
Total	188	2188	13301	5.6	6.1

Table 7: LitEval dataset statistics.

E Ablation Studies

Tables 8 and 9 report the full ablation results for all evaluation metrics.

F Supplementary Evaluation Details

Evaluator backbones. We instantiate both GEMBA-DA_{ref} and LiTransProQA (V_w) with

Model	XCOMET	XCOMET-QE	V_w
Qwen3-4B	54.66 (+5.23)	55.20 (+7.35)	4.08 (+0.15)
Qwen3-8B	57.19 (+4.13)	56.31 (+5.58)	4.17 (+0.07)
Qwen3-14B	61.95 (+6.18)	60.08 (+7.57)	4.21 (+0.01)

Table 8: Model size study on LitEval en-zh Online-Origin ($\uparrow$). CoT mode, softmax parent selection, 20 rounds max. Best round by XCOMET-QE argmax. Δ = best – Round 1.

Setting	XCOMET	XCOMET-QE	V_w
Full	57.19 (+4.13)	56.31 (+5.58)	4.17 (+0.07)
No MQM	56.66 (+3.51)	55.55 (+4.75)	4.16 (+0.07)
No Signal	56.59 (+3.00)	55.07 (+3.84)	4.13 (+0.03)

Table 9: Three-level MQM ablation on LitEval en-zh TTS-Origin ($\uparrow$). CoT mode, softmax parent selection, Qwen3-8B, 20 rounds max. “Full” = MQM diagnosis + score direction; “No MQM” = no diagnosis, optimizer sees score direction only; “No Signal” = no diagnosis, no direction (blind optimization).

GPT-4o-mini. For the XCOMET family, we use XCOMET-XL as the shared backbone for both XCOMET and XCOMET-QE, evaluated in reference-based and reference-free QE modes, respectively.

XCOMET and XCOMET-QE. XCOMET is a unified learned MT evaluation metric that supports both reference-based evaluation and reference-free quality estimation (QE) within a single model family. In addition to a sentence-level quality score, it predicts localized error spans with severity labels, providing a more interpretable view of translation quality. In this work, we use XCOMET-XL as the common backbone for both XCOMET and XCOMET-QE.

LiTransProQA (V_w). LiTransProQA is a reference-free, LLM-based QA metric designed specifically for literary translation evaluation. Rather than relying on automatically generated questions, it uses a curated set of professionally motivated evaluation questions targeting literary devices, cultural understanding, tone and authorial voice, coherence, and related dimensions. The question inventory is developed from literary translation resources and then filtered through two stages: importance ratings from professional literary translators and sensitivity analysis to remove questions with weak discriminative power. The final metric uses 25 selected questions.

For each question, the evaluator answers *Yes*, *No*,

Setting	XCOMET-QE [†]	XCOMET (ref) [†]
Optimizer temperature $\tau=0.1, T_{\text{trans}}=0.0$		
0.0 [†]	57.03±0.43	57.74±0.52
0.2	57.14±0.47	57.72±1.21
0.4	57.16±0.82	57.81±0.71
0.6	57.49±0.67	57.99±0.89
0.8	57.53±0.39	58.02±0.28
Rewrite-selection temperature $T_{\text{opt}}=0.0, T_{\text{trans}}=0.0$		
0.01	56.75±0.40	57.31±1.18
0.03	56.80±0.57	57.61±0.79
0.1 [†]	57.03±0.43	57.74±0.52
0.3	57.08±0.89	57.83±0.88
1.0	56.70±0.47	57.71±0.80
Translation temperature $T_{\text{opt}}=0.0, \tau=0.1$		
0.0 [†]	57.03±0.43	57.74±0.52
0.2	57.52±0.25	57.99±0.47
0.4	58.16±0.39	58.23±0.31
0.6	58.97±0.49	59.16±0.18
0.8	59.52±0.41	59.71±0.32

Table 10: Temperature ablation for online adaption on LitEval en-zh (CoT mode, Qwen3-8B-FP8, 49 paragraphs, max. 20 rounds). XCOMET-QE is the score of the QE-selected best round, while XCOMET (ref) is the reference-based score of the same round. All values are $\times 100$ and reported as mean $\pm$ std over three search seeds. [†] denotes the shared control run ($T_{\text{opt}}=0.0, \tau=0.1, T_{\text{trans}}=0.0$).

or *Maybe*, which are mapped to scores of 1, 0, and 0.5, respectively. The final score is computed by averaging across questions. In the weighted variant V_w used in this paper, question weights are derived from professional translator ratings. We report LiTransProQA on LitEval as a literary-focused auxiliary evaluator complementary to XCOMET-QE.

G Temperature Sensitivity Analysis

We examine the sensitivity of STEINS to three temperature parameters: the optimizer decoding temperature T_{opt} , the rewrite-selection softmax temperature τ , and the translation decoding temperature T_{trans} . We vary one parameter at a time while fixing the others to the default setting $(T_{\text{opt}}, \tau, T_{\text{trans}}) = (0.0, 0.1, 0.0)$; diagnosis and review temperatures remain zero throughout. Each condition is repeated three times.

Table 10 reports online adaptation results under Origin initialization on LITEVAL-CORPUS en-zh. XCOMET-QE selects the best translation over at most 20 adaptation rounds, while reference-based XCOMET evaluates the selected output without participating in search. The rewrite-selection temperature has only a modest effect, with $\tau = 0.1$ –

Setting	XCOMET [†]	XCOMET-QE [†]
Optimizer temperature $\tau=0.1, T_{\text{trans}}=0.0$		
0.0 [†]	76.02±0.92	74.41±1.29
0.4	76.86±0.84	75.09±0.44
0.8	76.09±1.27	74.19±0.68
Rewrite-selection temperature $T_{\text{opt}}=0.0, T_{\text{trans}}=0.0$		
0.01	75.64±0.12	74.03±1.13
0.1 [†]	76.02±0.92	74.41±1.29
1.0	75.95±1.18	73.25±0.34
Translation temperature $T_{\text{opt}}=0.0, \tau=0.1$		
0.0 [†]	76.02±0.92	74.41±1.29
0.4	75.72±0.82	73.49±1.69
0.8	74.59±1.14	72.48±1.84

Table 11: Temperature ablation for offline training on WMT23 en-zh (CoT mode, Qwen3-8B-FP8, 170 train / 30 test sentences, 20 epochs). For each run the reported epoch is selected by the train criterion (highest train XCOMET; ties broken by the earliest epoch); values are test XCOMET and XCOMET-QE at that epoch, $\times 100$, mean $\pm$ std over three seeds. [†]: shared control run.

Method	XCOMET-QE	XCOMET
Zero-shot	72.96	75.56
Few-shot (3-shot)	71.50 (-1.46)	74.93 (-0.63)
Self-refine (3 iter.)	74.08 (+1.12)	76.53 (+0.97)
Original Single	73.10 (+0.14)	76.16 (+0.60)
STEINS Single	79.08 (+6.12)	79.15 (+3.59)

Table 12: Comparison of translation quality across methods. Values in parentheses indicate improvements over the zero-shot baseline.

0.3 performing similarly. Optimizer temperature is also relatively insensitive over the tested range. In contrast, translation temperature shows the clearest trend: increasing T_{trans} from 0.0 to 0.8 improves XCOMET-QE from 57.03 to 59.52 and XCOMET from 57.74 to 59.71. Under a maximum budget of 20 adaptation rounds, greater translation diversity therefore appears beneficial for best-of-round selection. Among the evaluated values, $T_{\text{trans}} = 0.8$ performs best.

Table 11 reports the corresponding offline sensitivity analysis on WMT23 en-zh (170 train / 30 test split). A moderate optimizer temperature of $T_{\text{opt}} = 0.4$ performs best, reaching 76.86 XCOMET, whereas varying τ provides no clear improvement over the default $\tau = 0.1$. Unlike online adaptation, increasing translation temperature degrades offline performance: $T_{\text{trans}} = 0.8$ reduces XCOMET from 76.02 to 74.59 and XCOMET-QE from 74.41 to 72.48. Overall, the results suggest a phase-dependent trade-off: stochastic translation benefits online best-of-round exploration, whereas

offline corpus-level evolution favors more stable translation outputs.

H Additional Baseline Comparisons

We further compare STEINS with few-shot prompting and iterative self-refinement on WMT23 en-zh using Qwen3-8B-FP8. For a controlled comparison, we use the single-agent setting, since both few-shot prompting and self-refinement operate without multi-agent coordination.

Few-shot prompting performs slightly below the zero-shot baseline, while three-step self-refinement yields modest improvements of +1.12 XCOMET-QE and +0.97 XCOMET. In comparison, STEINS Single improves over zero-shot by +6.12 XCOMET-QE and +3.59 XCOMET on the same test set. These approaches are also complementary: few-shot or self-refinement prompts can in principle serve as the initial specification for subsequent STEINS adaptation. The results here isolate the effect of applying STEINS from the original single-agent specification.

I Supplementary Experiment Results

This section provides supplementary experimental results for online instance-level adaptation. Tables 14, 15, 16, and 17 report full per-setting results across datasets and initializations; Table 18 reports paired significance tests; Table 19 reports pairwise policy differences between Softmax and Sequential.

J AI Assistance Usage

We used AI-assisted tools, including ChatGPT², and Claude³, to support code development and writing refinement (in accordance with ACL Policy on AI Writing Assistance).

²<https://chatgpt.com>

³<https://claude.ai>



Figure 4: An end-to-end online adaptation search example for an English-Chinese instance. Round 2 executes s_1 , yielding $q(s_1) = 0.5692$; the diagnosis identifies an inaccurate rendering of "nursery fire" and redundant phrasing, and the optimizer creates s_2 with search credit $f_2 \leftarrow q(s_1)$. Round 3 executes s_2 , yielding $q(s_2) = 0.6392$, after which softmax rewrite selection samples the historical node s_1 with $P(s_1) = 0.5615$. Using the Round 3 diagnostic evidence, the optimizer rewrites s_1 to create the sibling child s_3 , assigning $f_3 \leftarrow q(s_2)$ as its search credit for subsequent archive sampling rather than as an independent evaluation of s_3 . Here c_i counts prior selections of s_i by the softmax policy and is initialized to zero, hence $c_1 = c_2 = 0$ at the first nontrivial softmax selection.

Mode	R=1	R=3	R=5	R=7	R=9	R=11	R=13	R=15	R=17	R=19	R=20	$\Delta(R20-R1)$
single	73.10	76.23	77.53	78.30	78.67	78.85	78.89	78.99	78.99	79.07	79.08	+5.99
CoT	72.76	76.61	77.91	78.37	78.75	79.04	79.16	79.24	79.33	79.33	79.33	+6.57
fixed	74.93	77.94	78.99	79.45	79.56	79.60	79.65	79.67	79.68	79.71	79.72	+4.79
MAD	71.50	74.34	75.39	75.79	76.03	76.20	76.25	76.29	76.35	76.37	76.37	+4.87

Table 13: Best-so-far XCOMET-QE ($\times 100$) across online adaptation rounds (en-zh, Qwen3-8B-FP8), averaged over 200 held-out sentences. Sentences that converge before round 20 hold their final value.

		Sequential			Softmax		
Mode	Lang	XCOMET	XCOMET-QE	GEMBA	XCOMET	XCOMET-QE	GEMBA
Qwen3-8B							
CoT	en-zh	79.08 (+3.84)	79.33 (+6.57)	87.28 (+0.12)	80.01 (+4.73)	80.47 (+7.71)	87.58 (+0.42)
	en-de	80.02 (+3.50)	81.34 (+5.19)	82.53 (+1.95)	80.71 (+4.16)	82.03 (+5.92)	82.83 (+2.35)
	de-en	87.54 (+1.73)	89.63 (+1.89)	86.97 (+0.40)	87.53 (+1.72)	89.73 (+2.00)	87.15 (+0.68)
Single	en-zh	79.15 (+2.99)	79.08 (+5.99)	87.53 (+0.30)	80.03 (+3.87)	80.04 (+6.97)	87.53 (+0.25)
	en-de	80.37 (+3.68)	81.31 (+5.14)	82.92 (+2.38)	80.22 (+3.56)	81.76 (+5.67)	82.78 (+2.25)
	de-en	87.84 (+1.40)	89.83 (+1.48)	87.00 (+0.58)	87.59 (+1.15)	89.99 (+1.63)	87.25 (+0.75)
Fixed	en-zh	79.57 (+2.97)	79.72 (+4.79)	87.30 (+0.47)	80.30 (+3.57)	80.56 (+5.48)	87.53 (+0.55)
	en-de	80.37 (+2.59)	81.48 (+3.67)	83.05 (+1.10)	80.60 (+2.99)	81.81 (+4.10)	82.95 (+0.90)
	de-en	87.85 (+0.84)	89.62 (+1.10)	87.40 (-0.05)	88.14 (+0.98)	89.95 (+1.46)	87.45 (-0.10)
MAD	en-zh	77.73 (+2.79)	76.37 (+4.87)	87.61 (+0.25)	78.17 (+3.09)	77.36 (+5.76)	87.79 (+0.43)
	en-de	79.91 (+3.20)	80.74 (+4.35)	82.37 (+0.63)	79.77 (+3.03)	80.94 (+4.52)	82.50 (+0.73)
	de-en	86.94 (+0.58)	88.68 (+1.10)	87.35 (+0.48)	86.91 (+0.54)	88.73 (+1.10)	87.06 (+0.13)
GPT-4o-mini							
CoT	en-zh	76.80 (+3.82)	76.54 (+6.23)	87.70 (+0.23)	78.42 (+2.72)	76.11 (+4.22)	88.22 (+0.12)
	en-de	82.73 (+1.33)	82.88 (+1.91)	85.75 (+0.30)	83.03 (+1.54)	82.99 (+2.00)	85.72 (+0.20)
	de-en	88.47 (+0.61)	89.75 (+0.95)	88.15 (-0.02)	88.53 (+0.74)	89.82 (+1.05)	88.40 (+0.38)
Single	en-zh	77.59 (+2.27)	75.54 (+4.50)	87.95 (+0.17)	78.38 (+3.11)	76.29 (+5.57)	88.28 (+0.43)
	en-de	82.63 (+1.36)	82.72 (+2.27)	85.70 (+0.43)	83.19 (+2.14)	82.96 (+2.54)	85.67 (+0.45)
	de-en	88.59 (+0.80)	89.80 (+1.04)	88.17 (-0.03)	88.64 (+0.84)	89.84 (+1.09)	88.60 (+0.47)
Fixed	en-zh	79.57 (+2.22)	77.85 (+3.84)	88.17 (+0.30)	79.97 (+2.70)	78.19 (+4.51)	88.15 (+0.15)
	en-de	83.79 (+1.19)	84.32 (+1.93)	85.93 (+0.23)	84.03 (+1.13)	84.34 (+2.05)	85.90 (+0.18)
	de-en	88.77 (+0.78)	90.14 (+1.33)	88.67 (-0.03)	88.73 (+0.69)	90.25 (+1.48)	88.70 (+0.10)
MAD	en-zh	78.69 (+2.19)	76.90 (+4.55)	88.33 (+0.10)	78.46 (+2.46)	77.07 (+5.22)	88.35 (+0.15)
	en-de	82.69 (+0.76)	83.02 (+1.59)	85.97 (+0.32)	82.69 (+0.91)	83.26 (+2.15)	85.90 (+0.25)
	de-en	88.71 (+0.76)	89.96 (+1.34)	88.62 (+0.08)	88.81 (+0.93)	90.11 (+1.34)	88.25 (-0.33)

Table 14: Online instance-level adaptation results on WMT23 (segment-level) under **Origin** initialization. Each cell reports the final selected score and its delta relative to the Round 1 baseline.

		Sequential			Softmax		
Mode	Lang	XCOMET	XCOMET-QE	GEMBA	XCOMET	XCOMET-QE	GEMBA
Qwen3-8B							
CoT	en-zh	79.23 (+3.69)	78.97 (+6.04)	87.33 (-0.05)	80.15 (+4.75)	80.29 (+7.74)	87.70 (+0.40)
	en-de	80.16 (+3.61)	81.30 (+5.16)	82.30 (+1.77)	79.21 (+2.77)	80.77 (+4.26)	82.35 (+1.35)
	de-en	87.44 (+1.43)	89.60 (+1.84)	87.12 (+0.53)	87.39 (+1.01)	89.94 (+1.50)	87.17 (+0.10)
Single	en-zh	79.33 (+2.67)	78.29 (+4.50)	87.28 (+0.00)	79.02 (+2.26)	78.77 (+5.06)	87.55 (+0.05)
	en-de	80.27 (+3.35)	81.11 (+4.72)	82.33 (+1.85)	80.02 (+2.61)	81.07 (+3.58)	82.12 (+1.67)
	de-en	87.34 (+0.67)	89.28 (+1.10)	86.95 (+0.15)	86.91 (+0.26)	88.65 (+0.41)	87.17 (+0.05)
Fixed	en-zh	79.47 (+2.82)	79.65 (+4.87)	87.25 (+0.28)	79.73 (+2.93)	79.98 (+4.90)	87.45 (+0.50)
	en-de	79.96 (+2.06)	80.43 (+2.91)	82.08 (+0.33)	80.85 (+2.97)	82.42 (+4.41)	82.75 (+0.85)
	de-en	88.01 (+1.03)	89.73 (+1.24)	87.28 (-0.15)	87.88 (+0.93)	89.81 (+1.64)	87.30 (+0.25)
MAD	en-zh	76.95 (+1.70)	76.02 (+3.12)	87.85 (+0.12)	78.90 (+3.21)	78.68 (+5.46)	87.72 (+0.15)
	en-de	79.73 (+1.32)	80.90 (+2.41)	82.83 (+0.84)	80.52 (+2.57)	82.05 (+4.15)	83.07 (+1.24)
	de-en	87.48 (+0.83)	89.56 (+0.98)	87.34 (+0.17)	87.40 (+0.72)	89.67 (+1.12)	87.22 (+0.15)
GPT-4o-mini							
CoT	en-zh	78.46 (+2.39)	75.61 (+3.66)	88.38 (+0.08)	77.79 (+2.14)	75.35 (+3.62)	88.28 (+0.18)
	en-de	83.01 (+1.19)	82.75 (+1.57)	85.47 (-0.10)	82.74 (+1.45)	82.81 (+2.05)	85.83 (+0.20)
	de-en	88.09 (+0.38)	89.39 (+0.94)	88.67 (-0.05)	88.49 (+0.65)	89.61 (+0.94)	88.62 (+0.25)
Single	en-zh	77.99 (+2.68)	75.56 (+5.12)	88.08 (+0.35)	77.01 (+1.05)	74.20 (+2.36)	88.17 (+0.33)
	en-de	82.25 (+0.47)	82.20 (+1.33)	85.35 (+0.10)	82.72 (+0.60)	82.73 (+1.71)	85.55 (+0.22)
	de-en	88.44 (+0.45)	89.41 (+0.78)	88.40 (-0.10)	88.55 (+0.61)	89.55 (+1.07)	88.47 (-0.23)
Fixed	en-zh	79.18 (+1.95)	77.24 (+3.80)	88.17 (+0.00)	79.48 (+3.38)	78.23 (+5.44)	88.17 (+0.33)
	en-de	83.98 (+1.11)	84.32 (+1.96)	86.03 (+0.18)	83.96 (+0.93)	84.41 (+2.21)	85.78 (-0.05)
	de-en	88.10 (+0.83)	89.60 (+1.29)	88.65 (+0.25)	88.70 (+0.59)	90.11 (+1.47)	88.60 (+0.12)
MAD	en-zh	79.39 (+2.49)	76.88 (+4.32)	88.15 (-0.15)	80.38 (+2.78)	78.46 (+5.25)	88.33 (+0.12)
	en-de	83.00 (+0.61)	83.50 (+1.95)	85.60 (+0.10)	83.32 (+1.16)	83.61 (+2.07)	85.42 (-0.17)
	de-en	88.43 (+0.78)	89.87 (+1.47)	88.78 (+0.28)	88.61 (+0.47)	89.97 (+1.03)	88.55 (+0.15)

Table 15: Online instance-level adaptation results on WMT23 (segment-level) under **Trained** initialization. Each cell reports the final selected score and its delta relative to the Round 1 baseline.

Mode	Lang	Sequential			Softmax		
		XCOMET	XCOMET-QE	V_w	XCOMET	XCOMET-QE	V_w

<i>Qwen3-8B</i>							
CoT	en-zh	56.52 (+3.40)	55.88 (+4.96)	4.15 (+0.06)	57.19 (+4.13)	56.31 (+5.58)	4.17 (+0.07)
	de-en	50.84 (+1.17)	57.31 (+4.37)	3.84 (-0.05)	52.11 (+2.43)	57.48 (+4.42)	3.90 (+0.02)
	en-de	65.11 (+3.62)	69.92 (+5.65)	3.76 (+0.14)	65.52 (+4.39)	70.34 (+6.55)	3.83 (+0.25)
Single	en-zh	57.66 (+3.98)	57.19 (+5.84)	4.15 (+0.03)	58.75 (+5.11)	57.30 (+6.00)	4.16 (+0.04)
	de-en	49.93 (+2.65)	56.05 (+5.21)	3.87 (+0.12)	51.01 (+3.85)	56.72 (+5.93)	3.87 (+0.13)
	en-de	64.58 (+4.71)	70.02 (+6.02)	3.83 (+0.21)	64.93 (+5.05)	70.12 (+6.11)	3.85 (+0.21)
Fixed	en-zh	58.73 (+4.12)	56.60 (+4.50)	4.16 (+0.05)	59.34 (+4.46)	58.84 (+7.07)	4.18 (+0.09)
	de-en	49.94 (+1.42)	55.57 (+3.80)	3.90 (-0.02)	51.58 (+2.86)	56.62 (+4.65)	3.89 (-0.02)
	en-de	65.73 (+3.40)	69.68 (+3.38)	3.90 (+0.12)	65.84 (+3.23)	71.10 (+4.66)	3.89 (+0.12)
MAD	en-zh	55.58 (+2.22)	55.28 (+4.57)	4.16 (+0.04)	56.01 (+2.63)	54.97 (+4.10)	4.16 (+0.05)
	de-en	47.90 (-0.27)	52.13 (+3.26)	3.78 (-0.02)	48.63 (+0.47)	52.44 (+3.57)	3.82 (+0.02)
	en-de	62.53 (+2.87)	69.14 (+4.92)	3.79 (+0.09)	62.80 (+3.32)	69.25 (+5.06)	3.78 (+0.07)

<i>GPT-4o-mini</i>							
CoT	en-zh	52.78 (+2.58)	52.25 (+3.85)	4.09 (+0.02)	52.37 (+2.64)	52.91 (+4.53)	4.13 (+0.07)
	de-en	51.68 (+1.61)	57.14 (+3.63)	3.90 (+0.05)	52.53 (+1.77)	57.48 (+3.34)	3.90 (+0.03)
	en-de	67.82 (+1.45)	71.61 (+2.89)	4.09 (+0.04)	68.24 (+2.58)	71.37 (+2.89)	4.06 (+0.03)
Single	en-zh	55.11 (+5.54)	52.73 (+4.95)	4.14 (+0.13)	54.19 (+3.97)	52.91 (+4.79)	4.16 (+0.10)
	de-en	51.81 (+2.62)	56.80 (+3.77)	3.93 (+0.05)	51.20 (+1.93)	57.03 (+4.50)	3.94 (+0.08)
	en-de	68.90 (+2.69)	71.61 (+3.82)	4.08 (+0.04)	68.02 (+1.88)	70.96 (+2.69)	4.09 (+0.04)
Fixed	en-zh	56.32 (+3.47)	55.89 (+4.97)	4.18 (+0.04)	56.12 (+3.31)	55.54 (+4.82)	4.16 (+0.03)
	de-en	52.03 (+3.50)	58.03 (+6.10)	4.05 (+0.03)	52.52 (+3.96)	57.70 (+5.70)	4.00 (-0.02)
	en-de	69.19 (+1.17)	72.40 (+3.90)	4.12 (+0.04)	70.85 (+3.44)	72.50 (+3.27)	4.12 (+0.03)
MAD	en-zh	53.67 (+2.57)	52.99 (+3.30)	4.10 (+0.04)	53.40 (+2.08)	52.49 (+3.19)	4.10 (-0.00)
	de-en	50.43 (+1.97)	56.47 (+3.75)	3.94 (+0.02)	50.86 (+2.40)	57.58 (+4.43)	3.93 (-0.02)
	en-de	67.36 (+2.07)	70.41 (+2.58)	4.06 (-0.01)	66.23 (+0.88)	70.36 (+1.75)	4.08 (+0.03)

Table 16: Online instance-level adaptation results on LitEval (paragraph-level) under **Origin** initialization. Each cell reports the final selected score and its delta relative to the Round 1 baseline.

Mode	Lang	Sequential			Softmax		
		XCOMET	XCOMET-QE	V_w	XCOMET	XCOMET-QE	V_w
CoT	en-zh	55.83 (+3.05)	56.08 (+5.24)	4.14 (+0.05)	57.72 (+4.45)	57.78 (+6.68)	4.16 (+0.07)
	de-en	51.67 (+2.07)	57.55 (+4.41)	3.85 (-0.02)	51.58 (+2.11)	57.16 (+4.92)	3.89 (+0.05)
	en-de	65.24 (+3.81)	70.38 (+6.19)	3.80 (+0.18)	64.17 (+4.66)	69.33 (+4.91)	3.78 (+0.21)
Single	en-zh	57.13 (+4.59)	56.34 (+6.96)	4.13 (+0.04)	57.63 (+3.99)	57.05 (+5.55)	4.15 (+0.09)
	de-en	50.76 (+2.61)	55.89 (+3.84)	3.82 (+0.04)	50.19 (+1.74)	54.92 (+3.07)	3.88 (+0.04)
	en-de	64.22 (+3.47)	69.49 (+4.59)	3.82 (+0.15)	64.94 (+2.65)	69.97 (+4.98)	3.77 (+0.09)
Fixed	en-zh	58.57 (+2.46)	57.00 (+3.63)	4.15 (+0.03)	58.54 (+2.72)	58.12 (+5.27)	4.15 (+0.03)
	de-en	50.03 (+1.40)	55.60 (+3.87)	3.88 (-0.03)	51.04 (+1.29)	56.17 (+3.77)	3.89 (-0.01)
	en-de	64.02 (+0.71)	69.43 (+3.04)	3.85 (+0.07)	65.35 (+2.72)	70.94 (+4.36)	3.88 (+0.10)
MAD	en-zh	56.24 (+2.18)	55.78 (+3.31)	4.17 (+0.00)	57.35 (+3.37)	56.70 (+4.29)	4.15 (-0.02)
	de-en	50.30 (+1.45)	55.53 (+2.75)	3.87 (+0.03)	50.81 (+2.46)	56.26 (+4.08)	3.85 (+0.01)
	en-de	63.06 (+2.24)	69.18 (+4.20)	3.79 (+0.07)	64.54 (+3.02)	70.02 (+4.78)	3.81 (+0.11)
<i>GPT-4o-mini</i>							
CoT	en-zh	53.22 (+2.16)	52.61 (+2.83)	4.10 (+0.04)	53.13 (+2.62)	52.97 (+3.61)	4.09 (+0.03)
	de-en	51.53 (+1.13)	56.85 (+3.55)	3.92 (+0.03)	51.83 (+0.69)	56.59 (+3.07)	3.87 (-0.02)
	en-de	68.13 (+2.07)	71.48 (+3.80)	4.11 (+0.04)	69.00 (+3.60)	71.57 (+3.36)	4.07 (+0.04)
Single	en-zh	53.00 (+2.95)	52.79 (+4.69)	4.10 (+0.07)	53.30 (+2.80)	52.48 (+4.38)	4.09 (+0.08)
	de-en	51.34 (+1.51)	56.11 (+2.13)	3.94 (-0.03)	51.70 (+0.62)	56.62 (+2.64)	3.91 (-0.03)
	en-de	69.68 (+3.60)	71.29 (+3.22)	4.04 (-0.04)	68.06 (+2.80)	71.00 (+2.66)	4.08 (-0.02)
Fixed	en-zh	56.04 (+2.64)	54.90 (+4.36)	4.16 (+0.02)	55.71 (+2.16)	54.35 (+3.88)	4.15 (+0.02)
	de-en	51.52 (+2.61)	57.15 (+5.74)	4.10 (-0.00)	50.89 (+1.16)	58.02 (+5.66)	3.98 (-0.05)
	en-de	69.92 (+2.13)	72.14 (+2.96)	4.05 (+0.01)	70.50 (+2.33)	72.36 (+3.63)	4.10 (+0.01)
MAD	en-zh	53.31 (+1.60)	52.15 (+2.87)	4.12 (+0.04)	54.83 (+2.86)	53.23 (+3.43)	4.15 (+0.06)
	de-en	51.41 (+2.31)	56.80 (+4.87)	3.96 (+0.00)	50.84 (+1.91)	56.60 (+3.88)	3.89 (-0.02)
	en-de	68.64 (+2.25)	71.85 (+3.82)	4.07 (-0.03)	67.55 (+0.71)	70.78 (+1.65)	4.09 (+0.02)

Table 17: Online instance-level adaptation results on LitEval (paragraph-level) under **Trained** initialization. Each cell reports the final selected score and its delta relative to the Round 1 baseline.

WMT23 Segment					
Init	Model	Policy	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta\text{GEMBA}\uparrow$
Origin	Qwen3-8B	Sequential	$+2.51 \pm 1.06$ (12/12)***	$+3.84 \pm 1.88$ (12/12)	$+0.72 \pm 0.72$ (11/12)***
		Softmax	$+2.78 \pm 1.30$ (12/12)***	$+4.36 \pm 2.19$ (12/12)	$+0.78 \pm 0.73$ (11/12)***
	GPT-4o-mini	Sequential	$+1.41 \pm 0.71$ (12/12)***	$+2.45 \pm 1.34$ (12/12)	$+0.19 \pm 0.17$ (9/12)***
		Softmax	$+1.66 \pm 0.87$ (12/12)***	$+2.77 \pm 1.58$ (12/12)	$+0.21 \pm 0.21$ (11/12)***
Trained	Qwen3-8B	Sequential	$+2.10 \pm 1.05$ (12/12)***	$+3.24 \pm 1.69$ (12/12)	$+0.49 \pm 0.64$ (9/12)***
		Softmax	$+2.25 \pm 1.23$ (12/12)***	$+3.69 \pm 2.05$ (12/12)	$+0.56 \pm 0.54$ (12/12)***
	GPT-4o-mini	Sequential	$+1.28 \pm 0.83$ (12/12)***	$+2.35 \pm 1.40$ (12/12)	$+0.08 \pm 0.16$ (7/12)
		Softmax	$+1.32 \pm 0.92$ (12/12)***	$+2.43 \pm 1.48$ (12/12)	$+0.12 \pm 0.17$ (9/12)*

LitEval Paragraph					
Init	Model	Policy	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta V_w \uparrow$
Origin	Qwen3-8B	Sequential	$+2.77 \pm 1.37$ (11/12)***	$+4.71 \pm 0.87$ (12/12)	$+0.06 \pm 0.07$ (9/12)***
		Softmax	$+3.49 \pm 1.25$ (12/12)***	$+5.31 \pm 1.02$ (12/12)	$+0.09 \pm 0.08$ (11/12)***
	GPT-4o-mini	Sequential	$+2.60 \pm 1.13$ (12/12)***	$+3.96 \pm 0.92$ (12/12)	$+0.04 \pm 0.03$ (11/12)***
		Softmax	$+2.57 \pm 0.91$ (12/12)***	$+3.83 \pm 1.09$ (12/12)	$+0.03 \pm 0.04$ (9/12)***
Trained	Qwen3-8B	Sequential	$+2.50 \pm 1.05$ (12/12)***	$+4.34 \pm 1.20$ (12/12)	$+0.05 \pm 0.06$ (10/12)***
		Softmax	$+2.93 \pm 0.99$ (12/12)***	$+4.72 \pm 0.88$ (12/12)	$+0.07 \pm 0.06$ (10/12)***
	GPT-4o-mini	Sequential	$+2.25 \pm 0.64$ (12/12)***	$+3.74 \pm 0.99$ (12/12)	$+0.01 \pm 0.03$ (8/12)
		Softmax	$+2.02 \pm 0.96$ (12/12)***	$+3.49 \pm 0.95$ (12/12)	$+0.01 \pm 0.04$ (7/12)

Table 18: Macro-averaged gains from online instance-level adaptation, measured as improvements of the final selected output over the Round-1 baseline across 12 settings (4 architectures $\times$ 3 language pairs), with paired t -test significance. * $p < 0.05$, ** $p < 0.01$, *** $p < 0.001$. $\Delta\text{xCOMET-QE}$, which serves as the optimization signal, is not significance-tested.

WMT23 Segment				
Init	Model	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta\text{GEMBA}\uparrow$
Origin	Qwen3-8B	$+0.27 \pm 0.39$ (7/12)	$+0.52 \pm 0.36$ (12/12)	$+0.06 \pm 0.21$ (7/12)
	GPT-4o-mini	$+0.25 \pm 0.29$ (10/12)	$+0.32 \pm 0.33$ (11/12)	$+0.02 \pm 0.26$ (6/12)
Trained	Qwen3-8B	$+0.15 \pm 0.78$ (5/12)	$+0.44 \pm 1.10$ (8/12)	$+0.08 \pm 0.31$ (7/12)
	GPT-4o-mini	$+0.04 \pm 0.68$ (7/12)	$+0.09 \pm 0.99$ (9/12)	$+0.04 \pm 0.21$ (6/12)

LitEval Paragraph				
Init	Model	$\Delta\text{XCOMET}\uparrow$	$\Delta\text{XCOMET-QE}\uparrow$	$\Delta V_w \uparrow$
Origin	Qwen3-8B	$+0.72 \pm 0.45$ (11/12)	$+0.60 \pm 0.75$ (11/12)	$+0.02 \pm 0.03$ (10/12)
	GPT-4o-mini	-0.03 ± 1.00 (6/12)	-0.13 ± 0.57 (4/12)	-0.01 ± 0.03 (4/12)
Trained	Qwen3-8B	$+0.43 \pm 0.89$ (8/12)	$+0.39 \pm 1.02$ (8/12)	$+0.01 \pm 0.03$ (7/12)
	GPT-4o-mini	-0.22 ± 0.91 (4/12)	-0.25 ± 0.80 (4/12)	$+0.00 \pm 0.03$ (8/12)

Table 19: Rewrite policy comparison of online adaptation (Softmax – Sequential). Values are mean $\pm$ std of per-setting Δ differences; parentheses show #settings where Softmax outperforms Sequential.

Error Category	Subtype	Description
Accuracy	Addition	Translation includes information not present in the source.
	Omission	Translation is missing content from the source.
	Mistranslation	Translation does not accurately represent the source.
	Untranslated text	Source text has been left untranslated.
	Other	Any other issues.
Fluency	Punctuation	Incorrect punctuation (for locale or style).
	Spelling	Incorrect spelling or capitalization.
	Grammar	Problems with grammar, other than orthography.
	Register	Wrong grammatical register (e.g., inappropriately informal pronouns).
	Inconsistency	Internal inconsistency (not related to terminology).
	Character encoding	Characters are garbled due to incorrect encoding.
	Other	Any other issues.
Terminology	Inappropriate for context	Terminology is non-standard or does not fit context.
	Inconsistent use	Terminology is used inconsistently.
	Other	Any other issues.
Style	Awkward	Translation has stylistic problems.
	Other	Any other issues.
Other Error	–	Any other issues.
Source error	–	An error in the source.
Non-translation	–	Impossible to reliably characterize distinct errors.

Table 20: MQM error hierarchy used in our diagnosis module.