U-MARL: Uncertainty-Aware Multi-Agent Reinforcement Learning for Word-Level MT Quality Estimation

Anonymous ACL submission

Abstract

Word-level machine translation (MT) quality estimation (QE) aims to localize erroneous spans in MT outputs. Despite the strong multilingual understanding of large language models (LLMs) and the collaborative capabilities of multi-agent systems (MAS), their token-level error judgments are often inherently uncertain or graded, whereas word-level QE requires deterministic OK/BAD labels, forcing such uncertainty to collapse into hard binary decisions. We argue that effective MAS-based word-level QE requires explicit uncertainty allocation, i.e., learning when to commit to an error hypothesis and when to defer uncertain cases for further adjudication. To this end, we propose U-MARL, a two-stage uncertainty-aware multi-agent reinforcement learning framework. U-MARL first trains a judge as a dedicated adjudication mechanism for deferred token-level cases, and then jointly optimizes error-specialist agents to learn selective adjudication policies under a QE-specific reward. Experiments on three language pairs show that, using open-source 4B non-reasoning LLMs as all agent backbones, U-MARL achieves competitive performance against closed-source-LLM-driven MAS baselines while reducing inference-time LLM calls. Our code is available at https://anonymous.4open.science/r/U-MARL-580F/.

1 Introduction

Machine translation (MT) quality estimation (QE) assesses the quality of MT outputs given the source sentence but without reference translations, making it essential for deployment scenarios where a system must decide whether a translation can be trusted, post-edited, or rejected (Specia et al., 2018; Zerva et al., 2024; Zhao et al., 2024). Unlike sentence-level QE, which provides only a coarse quality estimate for the whole translation (Specia et al., 2009; Kim et al., 2017; Ranasinghe et al., 2020), word-level QE assigns OK/BAD labels to individual target tokens, thereby localizing errors and providing actionable feedback for post-editing (Ueffing and Ney, 2007; Kreutzer et al., 2015; Kim et al., 2017). Learned automatic metrics such as COMET (Guerreiro et al., 2023) and MetricX (Juraska et al., 2024) have achieved strong performance in this task. However, they rely heavily on large amounts of human-annotated data and lack interpretability (Perrella et al., 2024; García Gilabert et al., 2025).

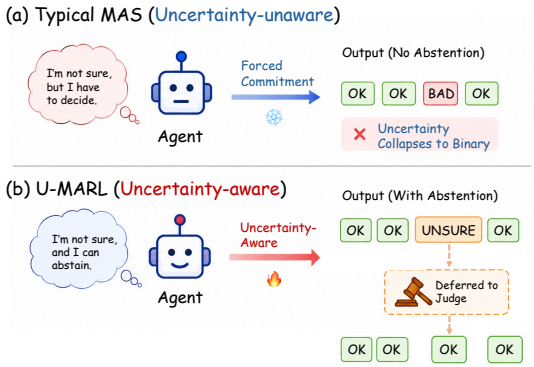

Figure 1: Comparison of (**a**) Typical MAS methods and (**b**) our U-MARL method.

Large language models (LLMs) have become a natural alternative for word-level QE, given their strong multilingual and translation capabilities (Zhu et al., 2024; Jiao et al., 2023; Anil et al., 2023; Hendy et al., 2023). However, prior works have shown that this potential does not necessarily lead to strong word-level QE performance (Huang et al., 2024; Lavie et al., 2025; Sarti et al., 2025; Kim, 2025). As illustrated in Figure 1, a key difficulty is that LLMs may produce inherently uncertain or graded judgments for ambiguous translation phenomena, whereas word-level QE requires these judgments to be expressed as deterministic binary labels, forcing model uncertainty to collapse into hard span decisions. For instance, the ambigu-

ity between subtle mistranslations and acceptable translation variants may leave the error status of an expression underdetermined, while vague span boundaries can leave token-level error attribution underspecified.

Recent multi-agent systems (MAS) for MT evaluation typically implement fixed collaborative schemes such as discussion, critique, or debate, indirectly reflecting and reconciling uncertain judgments (Zhang et al., 2025; Feng et al., 2025). Such inference-time mechanisms can surface disagreement, but they treat uncertainty only as a byproduct of interaction rather than as an explicit decision variable to be learned. Consequently, the system does not learn a QE-specific policy for deciding when a role-specific agent should finalize a token-level error hypothesis and when the hypothesis should be passed to subsequent resolution. We argue that improving MAS-based word-level QE therefore requires not only better error discovery, but also explicit uncertainty allocation across the multi-agent decision process.

Prior work on reliable LLMs suggests that models can be trained to improve uncertainty-awareness and avoid overconfident commitment through abstention or deferral (Zhou et al., 2026; Piatrashyn et al., 2026; Xue et al., 2025). This motivates us to operationalize uncertainty as an explicit trainable agent action in MAS-QE.

In this work, we propose U-MARL, an Uncertainty-aware Multi-Agent Reinforcement Learning framework for word-level QE. U-MARL operationalizes uncertainty through an explicit abstention action: role-specific agents issue OK/BAD decisions only for tokens supported by sufficient linguistic evidence, while uncertain tokens are deferred and routed to a judge through a selective-adjudication mechanism. We train the system in two stages. First, we supervise the judge as an adjudication operator that resolves deferred token-level cases. Second, we freeze the judge and optimize the role-specific agents with AT-GRPO (Zhao et al., 2026) under this fixed adjudication mechanism. The QE-specific reward jointly promotes accurate final predictions, role-consistent error detection, and appropriate abstention behavior, allowing the system to learn how decision authority should be allocated between specialized agents and the adjudicator.

We evaluate U-MARL on three language pairs, English→Marathi, Chinese→English, and English→German, using open-source 4B non-reasoning LLMs as all agent backbones. After two-stage training, U-MARL achieves competitive word-level QE performance against other MAS-QE baselines driven by GPT-5.5 (OpenAI, 2026), while reducing the LLM-call budget by at least 23%. Ablation studies further show that the explicit abstention mechanism, the trained judge, and the QE-specific reward each contribute to the final performance. Our main contributions are as follows:

- We propose an uncertainty-aware selective-adjudication workflow for multi-agent word-level QE that exposes uncertainty as an explicit token-level abstention action, allowing role-specific agents to defer uncertain cases to a judge.
- We introduce U-MARL, a two-stage MARL framework that casts uncertainty allocation as a policy-learning problem where specialist agents are optimized with a QE-specific reward that promotes accurate final prediction, useful role-level diagnosis, and calibrated abstention.
- Experiments on EN→MR, ZH→EN, and EN→DE show that U-MARL enables open-source 4B non-reasoning LLMs to achieve competitive word-level QE performance against GPT-5.5-driven MAS-QE baselines, while reducing the LLM-call budget by at least 23%.

2 Methodology

In this section, we first describe the uncertainty-aware selective-adjudication workflow (Section 2.1), then introduce how the judge agent is supervised to resolve deferred uncertain cases (Section 2.2), and finally present the joint reinforcement learning stage for optimizing the error-specialist agents (Section 2.3). Figure 2 gives an overview of the U-MARL framework.

2.1 Uncertainty-aware Selective-adjudication Workflow

U-MARL decomposes word-level QE into specialist-based candidate generation followed by selective adjudication. Given a source sentence and its machine translation, MQM-inspired specialists are assigned to linguistically motivated error categories and search for corresponding

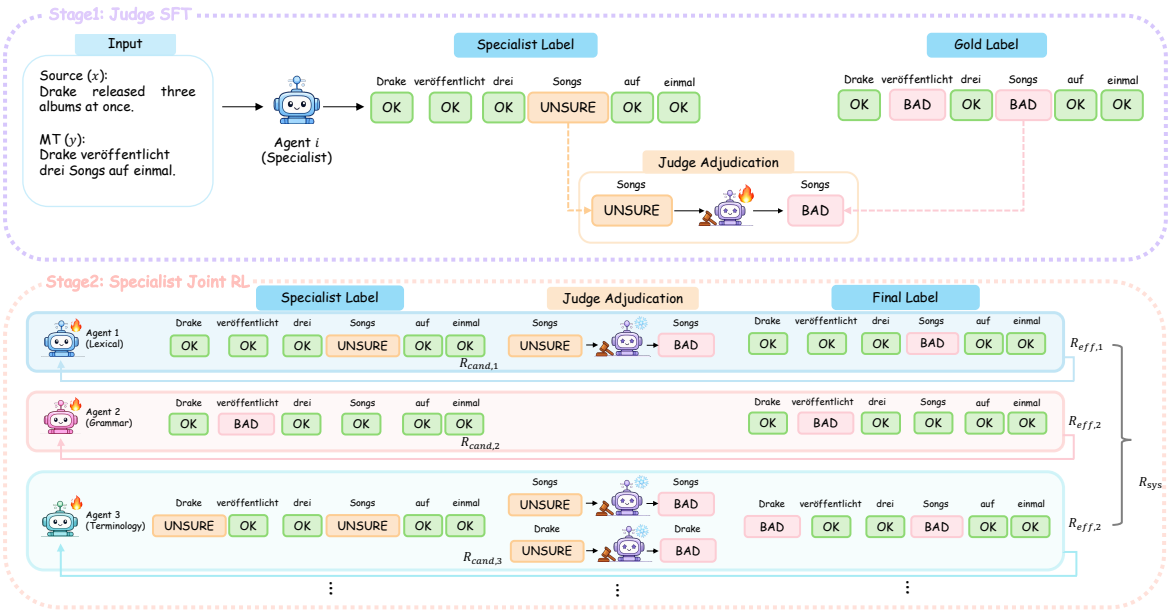

Figure 2: Overview of U-MARL for target-side word-level quality estimation. MQM-inspired specialists produce committed error spans or defer ambiguous cases as UNSURE; a supervised judge then keeps or drops deferred candidates before aggregation into token-level OK/BAD predictions. U-MARL first trains the judge and then freezes it while jointly optimizing the specialists with AT-GRPO under a QE-specific reward that promotes accurate prediction and calibrated uncertainty use.

target-side error spans within their respective scopes. Each span is either committed as an error and projected to token-level BAD predictions, or deferred as UNSURE when the available evidence is ambiguous or context-dependent.

The deferred UNSURE spans form uncertainty candidates that are passed to the judge for a KEEP/DROP decision. Only kept candidates are merged with directly committed specialist outputs and aggregated into the final BAD tokens. This workflow separates error discovery from final commitment: specialists maintain broad coverage over plausible error evidence, while the judge resolves uncertain cases to reduce spurious commitments.

2.2 Judge Supervised Fine-tuning

Before optimizing the MQM error-specialist agents, we first train the judge as a dedicated adjudication operator for deferred uncertain cases. We split the original training set into two partitions with a 1:4 ratio, and run the selective-adjudication workflow described in Section 2.1 on the 80% partition and collect all uncertainty candidates proposed by the error-specialist agents. Each candidate is represented as $c = (u^{src}, u^{mt}, t)$, where u^{src} and u^{mt} denote the corresponding source-side and target-side spans of the uncertainty candidate, and t is the gold token-level QE tag associated with u^{mt}. Let $z = (x, y, u^{src}, u^{mt})$ denote the judge input, where x is the source sentence and y is the machine translation. This process yields a judge-specific dataset $\mathcal{D}_J = \{(z, t)\}$.

After supervised fine-tuning, the judge is frozen and used as a fixed adjudication component in the subsequent reinforcement learning stage.

2.3 Joint Multi-agent Reinforcement Learning

We optimize the MQM error-specialist agents within the selective-adjudication workflow, using the remaining 20% partition of the training set. In this stage, policy updates are applied only to the error-specialist agents. The objective is to learn role-specific policies that balance committed error localization, uncertainty deferral, and non-redundant collaboration.

2.3.1 Notation

Let x be the source sentence, $y = (y_1, \ldots, y_n)$ the MT output, and $[n] = \{1, \ldots, n\}$. Given gold labels $g_k \in \{\text{OK}, \text{BAD}\}$, the gold error set is $G = \{k \in [n] : g_k = \text{BAD}\}$. Let $\mathcal{I}$ be the set of error-specialist agents and $\mathcal{I}_{\text{uns}} \subseteq \mathcal{I}$ the agents allowed to produce UNSURE candidates. For each $i \in \mathcal{I}_{\text{uns}}$, S_i denotes committed error items and U_i deferred uncertain items, with $U_i = \emptyset$ for $i \notin \mathcal{I}_{\text{uns}}$. Let J be the frozen judge and $\text{Tok}(\cdot)$ map items to

covered MT token positions. We define

$$K_i = \{u \in U_i : J(x, y, u) = \textsc{Keep}\},$$
$$E_i = \text{Tok}(S_i \cup K_i), \quad Q_i = \text{Tok}(S_i \cup U_i),$$

where K_i is the judge-kept uncertain set, E_i is the effective post-adjudication output, and Q_i is the pre-adjudication candidate set. The final predicted BAD-token set is $P = \text{Agg}(\bigcup_{i \in \mathcal{I}} E_i) \subseteq [n]$. We also define $V_i \in \{0, 1\}$ as whether agent i's output is parseable and all spans can be mapped to MT tokens.

2.3.2 Reward Function

All rewards are computed at the sentence level, treating BAD tokens as positive labels. For any predicted token set A and gold set G, we compute TP, FP, TN, and FN over the n MT tokens. We use MCC_ε, an ε-stabilized Matthews correlation coefficient to avoid division by zero of the standard MCC. We rescale it to $[0, 1]$ as

$$\overline{\text{MCC}}(A, G) = \frac{\text{MCC}_\varepsilon(A, G) + 1}{2}.$$

We use a common token-set scoring function

$$\mathcal{S}(A, G) = 0.6\,\overline{\text{MCC}}(A, G) + 0.4\,F_1(A, G).$$

This prioritizes MCC, the primary QE metric, while retaining a direct positive-class localization signal through F_1 under sparse BAD labels.

System reward. The global reward evaluates the final aggregated prediction as $R_{\text{sys}} = \mathcal{S}(P, G)$. This is a system-level trajectory reward shared by all error-specialist agents within the same rollout.

Effective contribution reward. To assign credit to each agent after adjudication, we compute $R_{\text{eff},i} = \mathcal{S}(E_i, G)$. This component rewards an agent for producing effective error evidence that survives the judge or is committed directly, while penalizing unsupported committed errors.

Candidate discovery reward. For agents equipped with the abstention mechanism, we additionally reward their ability to discover plausible error candidates before judge filtering: $R_{\text{cand},i} = \mathcal{S}(Q_i, G)$. Candidate discovery reward gives discovery credit when an agent identifies a genuine BAD token as either a committed error or an uncertain candidate. It therefore encourages specialists to expose ambiguous but potentially useful evidence, rather than suppressing such cases because of uncertainty.

Redundancy penalty. To discourage repeated cross-agent proposals, let $Q_{-i} = \bigcup_{j \in \mathcal{I}, j \neq i} Q_j$ and define

$$R_{\Delta,i} = \frac{|Q_i \cap Q_{-i}|}{\max(n, 1)}.$$

With a negative coefficient in the final reward, this term penalizes redundant predictions and encourages role-specific coverage.

Uncertainty-budget penalty. To prevent agents from overusing UNSURE as a risk-avoidance strategy, we impose a length-normalized uncertainty budget. Let

$$B_U(y) = \max\left(1, \left\lfloor \frac{n}{10} \right\rfloor\right)$$

be the allowed uncertainty budget for an MT sentence with n target tokens. For agents equipped with the abstention mechanism, we define

$$R^U_{\text{budget},i} = \mathbf{1}_{i \in \mathcal{I}_{\text{uns}}} \max\left(0, \frac{|U_i| - B_U(y)}{B_U(y)}\right).$$

This penalty discourages agents from deferring excessive numbers of candidates to the judge, ensuring that UNSURE is used for genuinely ambiguous cases rather than as a default safe action.

The final reward for agent i is then

$$R_i = -(1 - V_i) + V_i \boldsymbol{\lambda}_i \mathbf{r}_i.$$

Here, $\mathbf{r}_i$ collects the five reward components above, and $\boldsymbol{\lambda}_i = (\alpha_i, \beta_i, \gamma_i, \delta_i, \eta_i)$ denotes the reward weights with $\delta_i < 0$ and $\eta_i < 0$. If the output is invalid, the agent receives a fixed penalty of -1 and obtains no task reward. If the output is valid, the agent receives the weighted combination of the system-level reward, effective contribution reward, candidate discovery reward, redundancy penalty, and uncertainty-budget penalty. For agents without abstention, the coefficients of the candidate-discovery reward and the uncertainty-budget penalty are set to zero. The coefficients are reported in Appendix C.

2.3.3 Joint Optimization with AT-GRPO

We optimize the error-specialist agents with AT-GRPO, an on-policy RL algorithm designed for LLM-based MAS (Zhao et al., 2026). Unlike standard GRPO (Shao et al., 2024), which forms relative-reward groups over homogeneous completions, AT-GRPO groups rollouts by agent and turn. This is important in our setting because different specialists receive different role prompts and produce different types of outputs, so their rewards should not be normalized against heterogeneous agent behaviors.

Model	System	ZH→EN 23		EN→MR 23		EN→MR 22		EN→DE 23		ALC↓ across language pairs
		MCC↑	F1↑	MCC↑	F1↑	MCC↑	F1↑	MCC↑	F1↑	
					Encoder-based learned metric					
	XCOMET-XL	0.2120	0.2536	0.1360	0.2423	0.1697	0.2716	0.2095	0.2541	–
					Single-LLM					
Qwen3-4B-Instruct-2507	–	0.0850	0.1978	0.0781	0.1937	0.0635	0.1939	0.1317	0.2019	**1.00**
GPT-5.5	–	0.2040	0.2643	0.1625	0.2121	0.1819	0.2238	0.2083	0.2596	**1.00**
					Inference-time MAS					
Qwen3-4B-Instruct-2507	M-MAD	0.1553	0.2100	0.0850	0.2005	0.0854	0.2056	0.1144	0.1869	14.99
	HiMATE	0.1114	0.1693	0.0860	0.2000	0.0671	0.1943	0.1003	0.1709	63.40
GPT-5.5	M-MAD	0.2129	0.2314	0.1727	0.2618	0.2005	0.2870	0.2188	0.2508	9.16
	HiMATE	0.2169	0.2349	0.1607	0.2482	0.2011	**0.2903**	**0.2191**	0.2463	36.99
					Trainable MAS					
Qwen3-4B-Instruct-2507	**U-MARL**	**0.2539**	**0.3004**	**0.1818**	**0.2710**	**0.2059**	0.2817	0.2117	**0.2718**	7.06

Table 1: Main word-level QE results. MCC is the primary metric; F1 and average ALC across available language pairs are reported as complementary metrics. U-MARL uses Qwen3-4B-Instruct-2507 as the backbone for all agents and is trained with the proposed two-stage procedure.

3 Experiments

3.1 Experimental Setup

Dataset. We use publicly released ZH→EN and EN→DE data from the WMT23 Quality Estimation Shared Tasks (Blain et al., 2023) and EN→MR data from both WMT22 (Zerva et al., 2022) and WMT23. We do not include the WMT22 ZH→EN and EN→DE data, since prior work has questioned the suitability of these MQM-derived annotations for exhaustive word-level QE, as they may yield incomplete or noisy token-level OK/BAD labels (Yang et al., 2023). We independently train a system for each language pair. Detailed dataset statistics are presented in Appendix A.

Baselines. We compare U-MARL with three categories of baselines that are suitable for target-side word-level QE.

1) **Encoder-based learned metric:** We use XCOMET-XL as a strong encoder-based baseline (Guerreiro et al., 2023). XCOMET-XL is the XL variant of xCOMET, an explainable COMET-family metric trained to predict translation quality while also identifying target-side error spans.

2) **Single-LLM:** We include single-LLM baselines, where one LLM directly predicts target-side error spans from the source sentence and MT output. We instantiate this setting with Qwen3-4B-Instruct-2507 (Yang et al., 2025) and GPT-5.5 (OpenAI, 2026).

3) **Multi-agent system:** We further compare with representative multi-agent MT evaluation systems, including M-MAD (Feng et al., 2025) and HiMATE (Zhang et al., 2025). For each system, we instantiate two backbone variants, driven by Qwen3-4B-Instruct-2507 and GPT-5.5, respectively.

Evaluation Method. Following the official WMT22 and WMT23 word-level QE evaluation protocol, we use Matthews correlation coefficient (MCC) as the primary metric and report F1 as a complementary metric, treating BAD as the positive class.

In addition, we report the average number of LLM calls (ALC) per test sample as a lightweight inference-budget indicator. This quantity counts the average number of LLM invocations required to produce one final word-level QE prediction, including specialists, judge, and debate/adjudication calls when applicable.

3.2 Main Results

As shown in Table 1, U-MARL demonstrates strong word-level QE performance across language pairs. Although existing inference-time MAS protocols bring improvements over their corresponding single-LLM baselines, the gains are relatively limited. In contrast, U-MARL substantially outperforms Qwen3-4B-driven MAS baselines, improving over M-MAD by 93.9% in average MCC and 40.1% in average F1, and over HiMATE by

133.9% and 53.2%, respectively. This suggests that the major performance gain comes from turning MAS-QE into a trainable policy-learning problem, where specialists learn QE-specific error localization, abstention, and delegation behaviors under the final token-level objective, rather than from multi-agent interaction alone. U-MARL is also competitive with GPT-5.5-driven MAS systems, and even achieves higher average scores than M-MAD by 6.0% in MCC and 9.1% in F1, and than HiMATE by 7.0% and 10.3%, respectively. It further surpasses the comparable-scale encoder-based XCOMET-XL baseline by 17.3% in average MCC and 10.1% in average F1.

U-MARL also achieves this performance with a lower inference budget. It requires only 7.06 LLM calls per sample on average, reducing the call budget by at least 23% compared with GPT-5.5-driven MAS systems. This suggests that modeling uncertainty awareness as a learnable policy is more effective than relying on fixed discussion, critique, or debate-style inference procedures.

4 Ablation and Analysis

4.1 Component Ablation

We conduct component ablations to assess the main design choices in U-MARL, as shown in Table 2. We evaluate training-free role decomposition (RD) and selective adjudication (SA), judge training (JT), multi-agent reinforcement learning (MARL), and their combination, JT+MARL. We also test two uncertainty-collapse variants that replace the trained judge with UNSURE-accept (UA) or UNSURE-deny (UD).

RD improves error coverage over the single-LLM baseline, increasing recall by 29.2% while also improving MCC and F1. This confirms that role-specific specialists expose more target-side errors, although the broader candidate set slightly lowers precision. Adding SA further improves MCC, F1, and recall while keeping precision nearly unchanged, indicating that adjudication preserves the coverage gain without introducing many additional false positives.

Training yields larger gains. JT mainly improves recall by recovering genuine errors from deferred candidates, while MARL improves the specialists' commit-or-abstain decisions under the full workflow. The full JT+MARL model achieves the best overall result, improving MCC over JT alone by 102.9% and yielding the best F1 and precision. By contrast, UA and UD both degrade performance: accepting all deferred candidates introduces false positives, whereas denying them misses genuine errors. Thus, selective adjudication must be learned rather than reduced to a deterministic rule.

Method	MCC	F1	Precision	Recall
Training-free				
Single-LLM	0.0635	0.1939	0.1500	0.2742
RD	0.0900	0.2064	0.1456	0.3542
RD + SA	0.0929	0.2092	0.1459	0.3697
U-MARL				
JT	0.1015	0.2140	0.1461	**0.3995**
MARL	0.1401	0.2529	0.2094	0.3162
JT + MARL (OURS)	**0.2059**	**0.2817**	**0.2376**	0.3459
Uncertainty collapse				
UA	0.1762	0.2350	0.1723	0.3686
UD	0.1894	0.2640	0.2481	0.2811

Table 2: Component ablation results. RD denotes role decomposition, SA denotes selective adjudication and JT denotes judge supervised fine-tuning. UA and UD replace the trained judge by always accepting or always denying uncertain candidates, respectively. Experiments are conducted on the EN→MR 22 test set.

4.2 Analysis of Selective Adjudication

We trace the selective-adjudication workflow before and after U-MARL training in Figure 3. The left panel shows that agents without RL training exhibit a pronounced over-commitment pattern, directly committing a substantial number of gold OK tokens as errors. After RL training, the allocation shifts toward learned deferral: the uncertainty-candidate pool expands, and gold BAD candidates routed to this pool increase from 2.3% to 14.8%. This indicates that the deferred candidates are not merely more frequent, but also more enriched with true errors.

The right panel further shows that the adjudication stage becomes more selective after RL. For gold BAD uncertainty candidates, the judge keep rate increases from 59.33% to 67.48%; for gold OK candidates, the drop rate increases from 45.71% to 63.84%. Combining candidate allocation with judge decisions, the effective recovered-TP contribution from uncertainty candidates increases from $2.3\% \times 59.33\%$ before training to $14.8\% \times 67.48\%$ after training, corresponding to an approximately $7.4\times$ increase. Similarly, the effective filtered-FP contribution increases from $12.3\% \times 45.71\%$ to $38.6\% \times 63.84\%$, yielding a $4.38\times$ increase. These results suggest that the increased use of UNSURE does not merely transfer

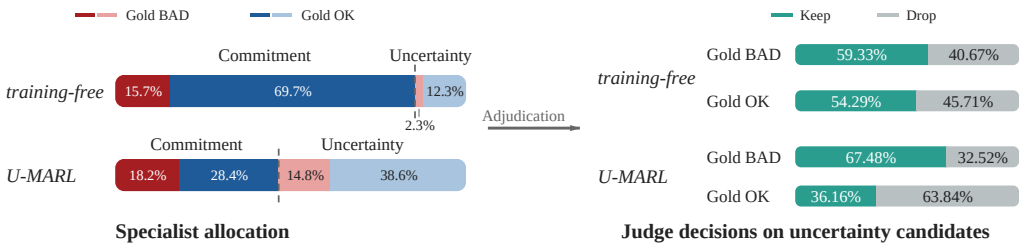

Figure 3: Flow analysis of selective adjudication. The left panel summarizes how training changes the allocation of direct commitments and deferred UNSURE candidates. The right panel illustrates how the judge keeps or drops deferred candidates conditioned on their gold labels. Experiments are conducted on the EN→MR 23 test set.

Reward	MCC	Uncertain-rate	Cand-Recall
only R_{sys}	0.1380	0.00	<u>0.305</u>
w/o R_{cand}	<u>0.1498</u>	<u>0.04</u>	0.235
OURS	**0.1818**	**2.00**	**0.393**

Table 3: Reward ablation study. *Uncertain-rate* measures the average number of UNSURE candidates generated per sample and *Cand-Recall* measures the recall of the pre-adjudication candidate set against gold BAD tokens. Bold indicates the best result and underline indicates the second best.

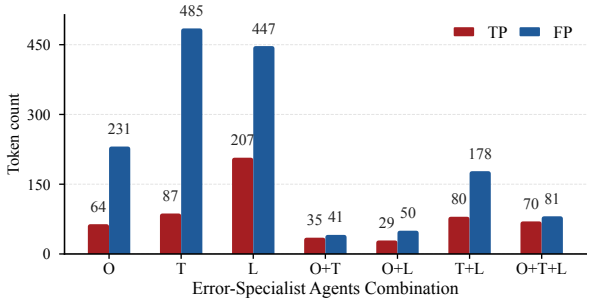

Figure 4: Role-wise overlap decomposition of U-MARL. The grouped bars compare exact specialist-overlap patterns for true-positive (TP) and false-positive (FP) candidates across the O (Omission), T (Terminology), and L (Loanword) specialists.

the decision burden to the judge; rather, it forms a larger and more informative adjudication channel that both recovers missed errors and suppresses false alarms.

4.3 Reward Shaping Enables Selective Adjudication

Table 3 studies how reward design affects the learned use of selective adjudication. With only the system-level reward R_{sys}, the model obtains an MCC of 0.1380 but produces no UNSURE candidates, showing that adding an abstention action alone still leads to hard decisions. Removing the candidate discovery reward, i.e., w/o R_{cand}, raises MCC to 0.1498, but the uncertainty rate remains near zero at 0.04 and candidate recall drops to 0.235, indicating little incentive to expose ambiguous candidates.

In contrast, the full reward achieves the best MCC of 0.1818 and increases the uncertainty rate and candidate recall to 2.00 and 0.393, respectively. This shows that R_{cand}, which rewards the pre-adjudication candidate set $Q_i = \text{Tok}(S_i \cup U_i)$, is essential for crediting plausible errors deferred through UNSURE. With this term, selective adjudication becomes an active policy component rather than an unused output option. The ablation therefore isolates candidate discovery as the component that turns abstention into selective routing.

4.4 Agent Contribution and Overlap

Figure 4 examines whether the specialists provide complementary evidence or redundant predictions, using Omission (O), Terminology (T), and Loanword (L) as representative roles. Most candidates fall into singleton rather than overlap regions. Taking exact pairwise-overlap regions as the baseline, singleton regions contain 148.6% more TP tokens and 332.3% more FP tokens. This suggests that U-MARL learns largely role-specific discoveries, with the redundancy penalty discouraging duplicated cross-agent proposals.

4.5 Case Study

Table 4 illustrates how U-MARL separates reliable error commitments from candidates requiring adjudication. Single-agent baselines either miss the error or select an incorrect span, while MAS-based baselines improve coverage but still make hard span decisions that may retain unsupported candidates.

U-MARL highlights the value of learned uncer-

Source	In Agartala, houses are being built by using New Zealand's steel frames technology. Such houses are considered better where the risk of earthquakes is high.
Translation	आगरतळा येथे न्यूझीलंडच्या पोलाद फ्रेम तंत्रज्ञानाचा वापर करून घरे बांधली जात आहेत, जिथे भूकंपाचा धोका जास्त असतो तिथे अशी घरे अधिक चांगली मानली जातात.
Annotation	पोलाद
Qwen3-4B-Instruct-2507	अशी घरे
GPT-5.5	Null
M-MAD$^{\triangle}$	आगरतळा,पोलाद
HiMATE$^{\triangle}$	आगरतळा,पोलाद
U-MARL*(training-free)	मानली जातात-sure,पोलाद-sure
U-MARL*(JT)	मानली जातात-sure,पोलाद-sure
U-MARL*(JT+RL)	आगरतळा-unsure-DROP,पोलाद-sure

Table 4: A case of responses generated by gpt-5.5 on MQM23 EN→MR among different LLM-based and MAS-based methods. The parts marked in red indicate incorrect judgments, while the parts marked in green indicate correct judgments. The superscript $\triangle$ indicates that the backbone is GPT-5.5, and $\star$ indicates that the backbone is Qwen3-4B-Instruct-2507.

tainty allocation. Before joint reinforcement learning, the system can discover the true error but still over-commits to spurious spans, which bypass the judge and cannot be corrected later. After optimization, U-MARL routes the ambiguous candidate through UNSURE, where the judge removes it, while preserving the genuine error prediction. This provides qualitative evidence that explicit abstention, selective adjudication, and the QE-specific RL objective improve prediction reliability.

5 Related Work

Machine Translation Quality Evaluation Machine translation quality evaluation has moved from reference-based overlap and edit-distance metrics such as BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), and TER (Snover et al., 2006) to reference-free QE models such as OpenKiwi (Kepler et al., 2019) and TransQuest (Ranasinghe et al., 2020). Learned evaluators including COMET (Rei et al., 2020), BLEURT (Sellam et al., 2020), CometKiwi (Rei et al., 2022), MetricX (Juraska et al., 2024), and xCOMET (Guerreiro et al., 2024) further improve human alignment and, in some cases, provide span-level evidence. Prompted LLM evaluators can be competitive at system-level judgment (Kocmi and Federmann, 2023), but they remain limited by calibration, consistency, low-resource generalization, reference dependence, span-level detection, and inference cost (Qian et al., 2024; Zerva et al., 2024; Lavie et al., 2025; Glushkova et al., 2021; Zerva and Martins, 2024).

Multi-agent Evaluation Multi-agent LLM-as-judge methods address the brittleness of single judges through role-based discussion and aggregation, as in ChatEval (Chan et al., 2024) and CourtEval (Kumar et al., 2025). For MT evaluation, M-MAD decomposes MQM-style assessment into debated dimensions (Feng et al., 2025), HiMATE organizes agents along the MQM hierarchy with reflection and asymmetric-information discussion (Zhang et al., 2025), and MAJ-EVAL similarly motivates task-grounded multidimensional personas (Chen et al., 2025). Yet most existing MAS evaluators remain prompt-orchestration frameworks that depend mainly on manually specified roles and debate protocols, inherit LLM-judge bias and calibration issues, and raise scalability and reproducibility concerns (Hashemi et al., 2024; Koo et al., 2024; Bavaresco et al., 2025; Qian et al., 2024; Lavie et al., 2025).

6 Conclusion

We proposed U-MARL, an uncertainty-aware multi-agent reinforcement learning framework for word-level QE, introducing UNSURE as an explicit trainable abstention action. Extensive experiments across three language pairs demonstrate that U-MARL enables open-source 4B non-reasoning LLM agents to achieve competitive word-level QE performance against GPT-5.5-driven MAS baselines with reduced LLM calls. We believe U-MARL moves multi-agent translation evaluation beyond fixed prompt orchestration, toward trainable and more reliable decision processes.

Limitations

To the best of our knowledge, the EN→MR datasets from WMT22 and WMT23 QE Shared Task are the only low-resource language-pair word-level QE resources that provide official train and test annotations, constraining the empirical scope of our study.

References

Rohan Anil, Andrew M. Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, Eric Chu, Jonathan H. Clark, Laurent El Shafey, Yanping Huang, Kathy Meier-Hellstern, Gaurav Mishra, Erica Moreira, Mark Omernick, Kevin Robinson, and 109 others. 2023. Palm 2 technical report. *Preprint*, arXiv:2305.10403.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, Andre Martins, Philipp Mondorf, Vera Neplenbroek, Sandro Pezzelle, Barbara Plank, David Schlangen, Alessandro Suglia, Aditya K Surikuchi, Ece Takmaz, and Alberto Testoni. 2025. LLMs instead of human judges? a large scale empirical study across 20 NLP evaluation tasks. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 238–255, Vienna, Austria. Association for Computational Linguistics.

Frederic Blain, Chrysoula Zerva, Ricardo Rei, Nuno M. Guerreiro, Diptesh Kanojia, José G. C. de Souza, Beatriz Silva, Tânia Vaz, Yan Jingxuan, Fatemeh Azadi, Constantin Orasan, and André Martins. 2023. Findings of the WMT 2023 shared task on quality estimation. In *Proceedings of the Eighth Conference on Machine Translation*, pages 629–653, Singapore. Association for Computational Linguistics.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2024. Chateval: Towards better LLM-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Jiaju Chen, Yuxuan Lu, Xiaojie Wang, Huimin Zeng, Jing Huang, Jiri Gesi, Ying Xu, Bingsheng Yao, and Dakuo Wang. 2025. Multi-agent-as-judge: Aligning llm-agent-based automated evaluation with multi-dimensional human evaluation. *Preprint*, arXiv:2507.21028.

Zhaopeng Feng, Jiayuan Su, Jiamei Zheng, Jiahan Ren, Yan Zhang, Jian Wu, Hongwei Wang, and Zuozhu Liu. 2025. M-MAD: Multidimensional multi-agent debate for advanced machine translation evaluation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7084–7107, Vienna, Austria. Association for Computational Linguistics.

Javier García Gilabert, Carlos Escolano, Audrey Mash, Xixian Liao, and Maite Melero. 2025. MT-LENS: An all-in-one toolkit for better machine translation evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (System Demonstrations)*, pages 51–60, Albuquerque, New Mexico. Association for Computational Linguistics.

Taisiya Glushkova, Chrysoula Zerva, Ricardo Rei, and {André F.T.} Martins. 2021. Uncertainty-aware machine translation evaluation. In *Findings of the Association for Computational Linguistics, Findings of ACL*, Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021, pages 3920–3938, United States. Association for Computational Linguistics (ACL). Publisher Copyright: © 2021 Association for Computational Linguistics.; 2021 Findings of the Association for Computational Linguistics, Findings of ACL: EMNLP 2021 ; Conference date: 07-11-2021 Through 11-11-2021.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xCOMET: Transparent machine translation evaluation through fine-grained error detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2023. xcomet: Transparent machine translation evaluation through fine-grained error detection. *Preprint*, arXiv:2310.10482.

Helia Hashemi, Jason Eisner, Corby Rosset, Benjamin Van Durme, and Chris Kedzie. 2024. LLM-rubric: A multidimensional, calibrated approach to automated evaluation of natural language texts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13806–13834, Bangkok, Thailand. Association for Computational Linguistics.

Amr Hendy, Mohamed Abdelrehim, Amr Sharaf, Vikas Raunak, Mohamed Gabr, Hitokazu Matsushita, Young Jin Kim, Mohamed Afify, and Hany Hassan Awadalla. 2023. How good are gpt models at machine translation? a comprehensive evaluation. *Preprint*, arXiv:2302.09210.

Xu Huang, Zhirui Zhang, Xiang Geng, Yichao Du, Jiajun Chen, and Shujian Huang. 2024. Lost in the

source language: How large language models evaluate the quality of machine translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3546–3562, Bangkok, Thailand. Association for Computational Linguistics.

Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Xing Wang, Shuming Shi, and Zhaopeng Tu. 2023. Is chatgpt a good translator? yes with gpt-4 as the engine. *Preprint*, arXiv:2301.08745.

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.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

Ahrii Kim. 2025. RUBRIC-MQM : Span-level LLM-as-judge in machine translation for high-end models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 147–165, Vienna, Austria. Association for Computational Linguistics.

Hyun Kim, Jong-Hyeok Lee, and Seung-Hoon Na. 2017. Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In *Proceedings of the Second Conference on Machine Translation*, pages 562–568, Copenhagen, Denmark. 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.

Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2024. Benchmarking cognitive biases in large language models as evaluators. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 517–545, Bangkok, Thailand. Association for Computational Linguistics.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. QUality estimation from ScraTCH (QUETCH): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 316–322, Lisbon, Portugal. Association for Computational Linguistics.

Sandeep Kumar, Abhijit A Nargund, and Vivek Sridhar. 2025. CourtEval: A courtroom-based multi-agent evaluation framework. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25875–25887, Vienna, Austria. Association for Computational Linguistics.

Alon Lavie, Greg Hanneman, Sweta Agrawal, Diptesh Kanojia, Chi-Kiu Lo, Vilém Zouhar, Frederic Blain, Chrysoula Zerva, Eleftherios Avramidis, Sourabh Deoghare, Archchana Sindhujan, Jiayi Wang, David Ifeoluwa Adelani, Brian Thompson, Tom Kocmi, Markus Freitag, and Daniel Deutsch. 2025. Findings of the WMT25 shared task on automated translation evaluation systems: Linguistic diversity is challenging and references still help. In *Proceedings of the Tenth Conference on Machine Translation*, pages 436–483, Suzhou, China. Association for Computational Linguistics.

OpenAI. 2026. GPT-5.5 System Card. https://openai.com/index/gpt-5-5-system-card/. Updated April 24, 2026; accessed 2026-05-25.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Stefano Perrella, Lorenzo Proietti, Pere-Lluís Huguet Cabot, Edoardo Barba, and Roberto Navigli. 2024. Beyond correlation: Interpretable evaluation of machine translation metrics. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20689–20714, Miami, Florida, USA. Association for Computational Linguistics.

Dzianis Piatrashyn, Nikita Kotelevskii, Kirill Grishchenkov, Nikita Glazkov, Ivan Nasonov, Ilya Makarov, Timothy Baldwin, Preslav Nakov, Roman Vashurin, and Maxim Panov. 2026. Redact: Uncertainty-aware deferral for llm agents. *Preprint*, arXiv:2604.07036.

Shenbin Qian, Archchana Sindhujan, Minnie Kabra, Diptesh Kanojia, Constantin Orasan, Tharindu Ranasinghe, and Fred Blain. 2024. What do large language models need for machine translation evaluation? In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3660–3674, Miami, Florida, USA. Association for Computational Linguistics.

Tharindu Ranasinghe, Constantin Orasan, and Ruslan Mitkov. 2020. Transquest: Translation quality estimation with cross-lingual transformers. *Preprint*, arXiv:2011.01536.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference*

on Empirical Methods in Natural Language Processing (EMNLP), pages 2685–2702, Online. Association for Computational Linguistics.

Ricardo Rei, Marcos Treviso, Nuno M. Guerreiro, Chrysoula Zerva, Ana C Farinha, Christine Maroti, José G. C. de Souza, Taisiya Glushkova, Duarte Alves, Luisa Coheur, Alon Lavie, and André F. T. Martins. 2022. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 634–645, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Gabriele Sarti, Vilém Zouhar, Malvina Nissim, and Arianna Bisazza. 2025. Unsupervised word-level quality estimation for machine translation through the lens of annotators (dis)agreement. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 18309–18326, Suzhou, China. Association for Computational Linguistics.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Matthew Snover, Bonnie Dorr, Rich Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of the 7th Conference of the Association for Machine Translation in the Americas: Technical Papers*, pages 223–231, Cambridge, Massachusetts, USA. Association for Machine Translation in the Americas.

Lucia Specia, Frédéric Blain, Varvara Logacheva, Ramón F. Astudillo, and André F. T. Martins. 2018. Findings of the WMT 2018 shared task on quality estimation. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 689–709, Belgium, Brussels. Association for Computational Linguistics.

Lucia Specia, Marco Turchi, Nicola Cancedda, Nello Cristianini, and Marc Dymetman. 2009. Estimating the sentence-level quality of machine translation systems. In *Proceedings of the 13th Annual Conference of the European Association for Machine Translation*, Barcelona, Spain. European Association for Machine Translation.

Nicola Ueffing and Hermann Ney. 2007. Word-level confidence estimation for machine translation. *Computational Linguistics*, 33(1):9–40.

Boyang Xue, Qi Zhu, Rui Wang, Sheng Wang, Hongru Wang, Minda Hu, Fei Mi, Yasheng Wang, Lifeng Shang, Qun Liu, and Kam-Fai Wong. 2025. Reliablemath: Benchmark of reliable mathematical reasoning on large language models. *Preprint*, arXiv:2507.03133.

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.

Zhen Yang, Fandong Meng, Yuanmeng Yan, and Jie Zhou. 2023. Rethinking the word-level quality estimation for machine translation from human judgement. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2012–2025, Toronto, Canada. Association for Computational Linguistics.

Chrysoula Zerva, Frederic Blain, José G. C. De Souza, Diptesh Kanojia, Sourabh Deoghare, Nuno M. Guerreiro, Giuseppe Attanasio, Ricardo Rei, Constantin Orasan, Matteo Negri, Marco Turchi, Rajen Chatterjee, Pushpak Bhattacharyya, Markus Freitag, and André Martins. 2024. Findings of the quality estimation shared task at WMT 2024: Are LLMs closing the gap in QE? In *Proceedings of the Ninth Conference on Machine Translation*, pages 82–109, Miami, Florida, USA. Association for Computational Linguistics.

Chrysoula Zerva, Frédéric Blain, Ricardo Rei, Piyawat Lertvittayakumjorn, José G. C. de Souza, Steffen Eger, Diptesh Kanojia, Duarte Alves, Constantin Orăsan, Marina Fomicheva, André F. T. Martins, and Lucia Specia. 2022. Findings of the WMT 2022 shared task on quality estimation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 69–99, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Chrysoula Zerva and André F. T. Martins. 2024. Conformalizing machine translation evaluation. *Transactions of the Association for Computational Linguistics*, 12:1460–1478.

Shijie Zhang, Renhao Li, Songsheng Wang, Philipp Koehn, Min Yang, and Derek F. Wong. 2025. HiMATE: A hierarchical multi-agent framework for machine translation evaluation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11121–11145, Suzhou, China. Association for Computational Linguistics.

Haofei Zhao, Yilun Liu, Shimin Tao, Weibin Meng, Yimeng Chen, Xiang Geng, Chang Su, Min Zhang, and Hao Yang. 2024. From handcrafted features to llms: A brief survey for machine translation quality estimation. In *2024 International Joint Conference on Neural Networks (IJCNN)*, page 1–10. IEEE.

Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. 2026. Stronger-MAS: Multi-agent reinforcement learning for collaborative LLMs. In *The Fourteenth International Conference on Learning Representations*.

Xinyu Zhou, Chang Jin, Carsten Eickhoff, Zhijiang Guo, and Seyed Ali Bahrainian. 2026. When silence is golden: Can llms learn to abstain in temporal qa and beyond? *Preprint*, arXiv:2602.04755.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2024. Multilingual machine translation with large language models: Empirical results and analysis. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2765–2781, Mexico City, Mexico. Association for Computational Linguistics.

A Datasets

Table 5 summarizes the WMT word-level QE datasets used in our experiments, covering EN→MR, ZH→EN, and EN→DE. We report the number of samples, the proportion of error-free examples, and the token-level BAD/OK label counts for each train or test split.

B Agent Role Assignment

For all language pairs, we implemented five error-specialist agents: Omission, Grammar, Factual Consistency, Terminology, and Loanword. This was based on a trade-off between available computational resources, time, and performance. More error-specialist agents can result in higher error coverage, but the time and computational resources required for training and inference will also increase accordingly.

C Reward Weights

We use the following base reward-weight vector:

$$\boldsymbol{\lambda} = (0.3, 0.2, 0.25, -0.1, -0.1)^{\top}.$$

The five entries correspond to the system reward, effective contribution reward, candidate discovery reward, redundancy penalty, and uncertainty-budget penalty, respectively. Agents without abstention use the same vector with the third and fifth entries set to 0. We note that $\boldsymbol{\lambda}$ was relatively sensitive in the experiments conducted in this study: small changes to these weights could noticeably affect the balance.

D Prompt Templates

For readability, we show two representative error-specialist prompt templates and the judge prompt template in Figure 5–7. The remaining error-specialist agents follow the same template structure, with minor modifications to the role description and error scope to match the corresponding error category.

Omission Specialist Prompt

You are an expert in detecting omissions in machine translation from {src_lang} to {tgt_lang}.
Your task is to determine whether the machine translation (MT) is missing any content from the source sentence and localize each omission in the MT.

IMPORTANT RULES:
1. Do NOT judge translation quality or fluency.
2. Do NOT assume missing information can be inferred from context.
3. A concept is considered PRESENT if it is explicitly expressed in the translation.
4. If a concept from the source sentence is not explicitly expressed, and the translation did not try to translate it, it must be reported as MISSING.
5. If the translation contains a word or phrase that attempts to express the same concept but the meaning is incorrect or incomplete, mark it as MISTRANSLATED, not MISSING.
6 Do NOT report omission for semantically light function words if their absence does not materially change the core propositional meaning of the sentence.
Examples include articles such as "a", "an", and "the".
However, do NOT ignore missing items that affect negation, quantity, comparison, modality, focus, referential contrast, or scope.
7. If the sentence is an imperative or similar, and the machine translation has adequately conveyed the meaning, then the "you" in the original sentence does not need to be explicitly translated.
8. Only mark MISSING when the concept does not appear anywhere in the translation.
9. For every unit marked as MISSING, you must also localize the omission in the MT, i.e., the position where it should be but is not present.
10. If multiple units considered "MISSING" are directly connected together, you can merge them and output them as one.
11. There are two different types of omission, each corresponding to a different localization method.
- local omission: A local omission happens when some source content is missing in the translation, but the surrounding translated structure remains compatible with the

Dataset	Split	Samples	Error-free (%)	Gold BAD tokens	Gold OK tokens
EN→MR 22	Test	1,000	36.60	2,064	19,047
EN→MR 23	Test	991	40.16	1,982	19,046
EN→MR	Train	4,333	9.99	9,023	84,710
ZH→EN 23	Test	1,664	66.65	2,999	41,239
ZH→EN 23	Train	4,333	9.99	20,166	15,687
EN→DE 23	Test	1,887	59.72	2,676	36,273
EN→DE 23	Train	4,333	9.99	14,606	150,039

Table 5: Dataset statistics for the WMT word-level QE data used in our experiments. Error-free denotes the proportion of sentence-level samples containing no gold BAD target tokens.

source. The error can be corrected mainly by inserting the missing content at an appropriate position, without rewriting the rest of the sentence.
- structural omission: A structural omission happens when the omitted source content is essential to the sentence's predicate-argument structure or core proposition. Because of the omission, the translation is no longer a partial rendering of the source but a different sentence meaning or structure. This type cannot be fixed by simple insertion; the affected span must be rewritten, so the whole span should be marked as erroneous.
12. When omissions are detected, apply the following two rules to assist in determining the type of omission.
Rule 1: Can the translation be fixed by insertion only?
If the missing source content can be added into the existing translation without substantially changing the surrounding words or structure, classify it as local omission.
Rule 2: Does the omission change the whole proposition or syntactic relation?
If the missing content is so central that its absence changes the predicate, event structure, or main semantic relation of the sentence, and the translation would need substantial rewriting rather than insertion, classify it as structural omission.
- NOTE: Prefer treating it as a local omission. Structural omission should be used only when the omission corrupts the internal semantic structure of the affected span so that simple insertion is insufficient and rewriting is necessary. When a sentence contains multiple clauses or half-clauses, omission in one clause does not automatically make the other clauses BAD.
13. For local omission, attempt to repair the translation by inserting the missing parts, and determine the location of the omission by comparing the sentences before and after the repair. Do not simply rely on {src_lang} word order to judge the location of the omission; you must remain faithful to {tgt_lang} grammar.
14. Always mark the SMALLEST contiguous span that is made incorrect by the omission. Do not expand the BAD span to the whole sentence unless the whole sentence must be rewritten.

Follow these steps strictly.
Step 1: Decompose the source sentence into semantic units.
You MUST extract units from ALL of the following categories:
1. Entities and objects
(people, things, places)
2. Events or states
(actions, relations, properties)
3. Modality, evidentiality, or speaker stance
Examples include:
- is known to
- is believed to
- is said to
- reportedly
- seems to
- appears to
- allegedly
- is considered to
These expressions MUST be treated as independent semantic units.

Step 2: For each unit, check whether the same meaning appears explicitly in the MT.
Use the following decision rules:
- If the translation clearly expresses the concept -> PRESENT
- If the translation contains a word or phrase of the same role or type that tries to express the concept but does it incorrectly -> MISTRANSLATED
- If the translation contains no expression corresponding to the concept at all -> MISSING

Step 3: Mark each unit as:
- PRESENT
- MISTRANSLATED
- MISSING

Step 4: Locate Omissions.
For each unit marked "MISSING", determine whether it belongs to "local omission" or " structural omission" based on its corresponding content in the machine translation.
- For local omission,
1. Obtain the correct translation of the smallest span surrounding the missing word.
2. Compare it with the current MT to identify the actual omission location.

3. The location marker is "<local>left_word [OMS] right_word</local>". If the omission occurs at the beginning of the sentence, only output "[OMS] right_word". If the omission occurs at the end, only output "[OMS] <EOS>".
- For structural omission, output the entire span affected by the omission in the translation. Strictly control span length; do NOT add any correctly translated words to the span. The location marker is <structural>span</structural>.

Output format:
Missing content: YES / NO
If YES, directly list the units (content, not serial numbers) that were determined to be MISSING, and strictly output them in the following format:
<miss>unit</miss><local>left_word [omission] right_word</local>
or
<miss>unit</miss><structural>span</structural>

Output should strictly follow the format, and do not output anything else.

Source:
{src_sentence}
Machine Translation:
{mt_sentence}

Figure 5: Prompt template for the omission specialist.

Grammar Specialist Prompt

You are an expert in detecting grammatical consistency errors in machine translation from {src_lang} to {tgt_lang}.
Your task is to determine whether the machine translation (MT) contains grammatical inconsistencies related only to:
1. Number consistency (singular/plural)
2. Tense consistency (past/present/future and obvious aspectual-temporal mismatches that affect tense interpretation)

You must check grammar from ONLY ONE perspective:
Source-to-Translation consistency
Determine whether the MT preserves number and tense information expressed in the source sentence.
Do NOT check internal consistency inside the MT.

IMPORTANT RULES:
1. Only evaluate number and tense consistency.
2. Do NOT evaluate fluency, style, adequacy, terminology, spelling, punctuation, or general grammar beyond number/tense.
3. Report errors only when they are clearly reflected in the MT.
4. Do NOT report a problem if the source does not explicitly encode the grammatical distinction.
5. Do NOT report lexical mistranslation unless it directly causes a number or tense inconsistency.
6. Prefer minimal error spans from the machine translation only.
7. Do NOT mark the whole sentence unless absolutely necessary.
8. When evidence is insufficient or ambiguous, do NOT report an error. Report an error ONLY if the MT contains an OVERT and UNAMBIGUOUS grammatical contradiction.
9. If the {tgt_lang} form is ambiguous, underspecified, syncretic, or compatible with the source, do NOT report an error.
10. Do NOT assume missing information can be inferred from context.
11. The span of errors must NOT be arbitrarily expanded; for each error, the error span should be minimized.
12. Each error contains only a single minimal failing part; combinations of incorrect parts with correct parts are prohibited.

Follow these steps strictly.
Step 1: Identify explicit number and tense cues in the source sentence.
Only consider cues that are clearly expressed.
Step 2: Compare the source cues with the MT.
Check whether number and tense are preserved.
Step 3: Report only actual grammatical errors in the MT.
Each reported error must:
- appear in the MT
- be a clear number or tense problem
- use the shortest possible error span

Output format:
If no error exists:
Grammatical error: NO
If one or more errors exist:
Grammatical error: YES
<error>...</error>
<error>...</error>

Source:
{src_sentence}
Machine Translation:
{mt_sentence}

Figure 6: Prompt template for the grammar specialist.

Judge Prompt

You are an expert for MT quality evaluation.

Task:
Only process potential error, i.e., <unsure> items, from other agents. Determine whether

the potential error is truly a translation error or not.
Do NOT evaluate the whole translation. Do NOT judge fluency, grammar, omission, or style. Do NOT find new errors.
Only focus on the current potential error.

Input:
1. Source sentence in {src_lang}
2. Machine translation (MT) in {tgt_lang}
3. Potential error from fact / terminology / loanword agents

For the potential error:
1. Read the claim.
2. Check whether the potential error is truly a mt error or not.
3. If you support the error, output RESOLVED_KEEP.
4. If you reject the error, output RESOLVED_DROP.

Output format:
<decision>RESOLVED_KEEP| RESOLVED_DROP</decision>

Only output the above one line.Strictly follow the format. Do not output any other content.

Source:
{src_sentence}

Machine Translation:
{mt_sentence}

Potential error:
{unsure_item}

Figure 7: Prompt template for the judge.

E Training Dynamics

Figure 8 summarizes the training dynamics for the EN→MR language pair. We distinguish two diagnostic stages in training: *Format Learning*, where the model mainly learns to follow the required output format so that the response can be parsed and mapped to MT tokens, and *Capability Learning*, where optimization focuses more directly on the QE behavior after most outputs have become parseable. The plotted reward curves correspond to the final agent reward R_i and the reward components defined in Section 2.3.2, including R_{sys}, $R_{\text{eff},i}$, $R_{\text{cand},i}$, $R_{\Delta,i}$, and $R^{U}_{\text{budget},i}$. Overall, R_i, R_{sys}, $R_{\text{eff},i}$, and $R_{\text{cand},i}$ increase during Capability Learning, while the format-error count drops sharply after the initial Format Learning stage and remains low thereafter. $R_{\Delta,i}$ and $R^{U}_{\text{budget},i}$ are noisier and should be interpreted as qualitative diagnostics of the optimization process rather than as primary evaluation metrics.

Figure 8: Training dynamics for the EN→MR language pair. Each panel is exported from W&B and reports one monitored signal during the self-defined Format Learning and Capability Learning stages.