

CONTEXTROUTER: Learning When to Use Context for Efficient LLM Machine Translation

Ritvik Choudhary Rem Hida Toshiyuki Sekiya

Sony Group Corporation

ritvik.choudhary@sony.com

Abstract

LLM-based machine translation systems can now do more than translate a source segment in isolation. At test time, they can translate directly, prepend previous context, or refine a context-aware draft. However, these interventions are not uniformly beneficial. Context can resolve discourse-sensitive choices, but irrelevant context and extra refinement increases cost and may perturb adequate translations. We introduce CONTEXTROUTER, a lightweight policy router that learns which intervention to apply for each source segment. During training, we first generate candidates for all contextual interventions and score them with a routing objective combining overall translation quality with incurred token costs. We then train our CONTEXTROUTER with group-relative supervision, so it learns within-example preferences rather than absolute global action scores. At inference, our router uses only test-time-available signals from the source, previous source-side context, and a no-context draft to determine and execute only the selected context intervention. On a filtered context-aware subset of TED, QED, and OpenSubtitles covering multiple languages and domains, our CONTEXTROUTER improves the quality-cost tradeoff over fixed no-context, always-context, always-refinement, QE-reranking, and score-regression policies, and human evaluation confirms that annotators prefer its chosen interventions. These results support treating context and refinement actions in LLM-MT not as always present defaults but conditional test-time interventions instead.

1 Introduction

Recent LLM-based machine translation (LLM-MT) is no longer limited to isolated sentence-level decoding. Given a source segment, the same frozen generator can either translate directly, translate with preceding context, or revise a draft translation (Zhang et al., 2023; Wang et al., 2023). This flexibility is useful because many translation deci-

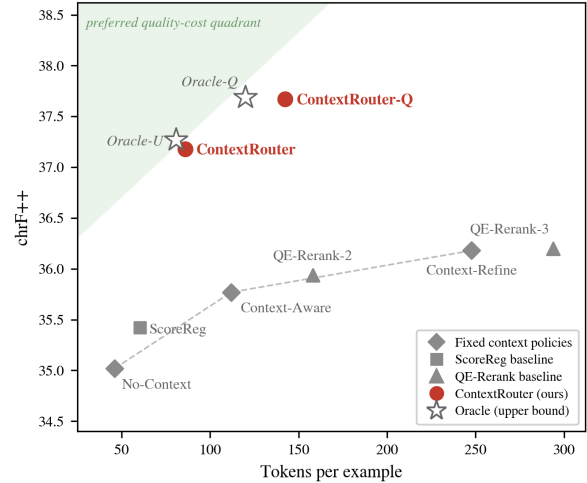


Figure 1: Quality-cost frontier on Qwen3.6-27B. The x -axis shows token cost per example; the y -axis shows chrF++. CONTEXTROUTER improves quality over fixed no-context and context-aware translation while remaining cheaper than always using context.

sions are discourse- or domain-sensitive where terminology, pronouns, lexical cohesion, and speaker intent may depend on earlier segments rather than on the current source segment alone (Choudhary et al., 2025; Bawden et al., 2018; Voita et al., 2019).

However, providing additional context is not a free input. Longer prompts increase inference cost, while irrelevant or misleading context can distract the model or induce fluent but unsupported translations (Sun et al., 2026; Castilho and Knowles, 2025; Rikters and Nakazawa, 2024). Self-refinement of translations has a similar tradeoff. Asking an LLM to translate again can repair an awkward draft (Feng et al., 2025; Wu et al., 2025; Chen et al., 2024b), but it also requires additional calls and may unnecessarily alter a translation that was already adequate (Gain et al., 2026; Ki and Carpuat, 2024). Thus, for LLM-MT the practical question is not whether context or refinement helps on average, but *when* each such intervention is worth applying.

We formulate this problem as *test-time intervention routing*. For each example in MT, the input is a source segment x with preceding source-

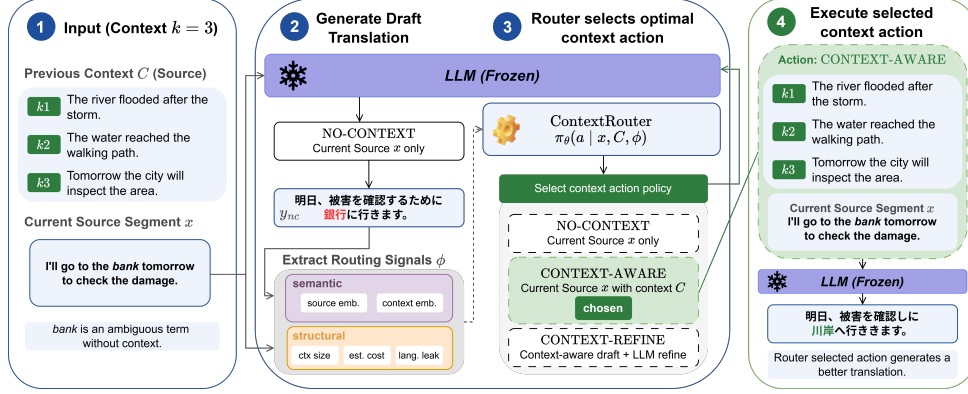


Figure 2: Overview of CONTEXTROUTER at inference. The LLM first produces a NO-CONTEXT draft; the router then uses semantic and structured signals to choose one contextual intervention: NO-CONTEXT, CONTEXT-AWARE, or CONTEXT-REFINE and finally only the selected intervention is executed.

side context $C = (c_1, \dots, c_k)$; the output is one target-language translation y . A translation intervention specifies how the generator uses x , C , as well as any intermediate draft. Following previous work, we consider three contextual interventions: NO-CONTEXT, which translates x directly without any context; CONTEXT-AWARE, which translates x with all available previous source-side context; and CONTEXT-REFINE, which generates a context-aware draft and then refines it using the source, context, and draft. Unlike context-aware MT, which focuses on representing or selecting discourse context (Maruf et al., 2019; Castilho and Knowles, 2025; Choudhary et al., 2025), LLM-MT prompting work that evaluates fixed pipelines (Vilar et al., 2023; Karpinska and Iyyer, 2023; Koneru et al., 2024), or model-routing systems that choose among models under resource constraints (Huang et al., 2025; Chen et al., 2024a; Luo et al., 2026), our setup keeps the generator fixed and learns an instance-level policy over the possible translation interventions. Therefore at each step the router decides whether adding context or performing an additional generation step is worth its cost for the current source segment.

We introduce CONTEXTROUTER, an offline group-relative route policy for this setting. During training, each example yields one candidate translation per contextual intervention, scored with a routing objective combining overall translation quality and token cost. Furthermore, taking inspiration from Shao et al. (2024); Garcia Gilabert et al. (2025), CONTEXTROUTER uses a group-relative target during training. Instead of training only on the single best action, candidate rewards are standardized within each example and converted into a soft distribution over interventions. This makes supervision comparable across language pairs and

domains whose reward scales differ widely, while a non-negligible normalization floor softens the target when actions are nearly tied, a common situation when context provides only small instance-specific gains. Then at test-time, the router observes only the signals available before generating any contextual intervention: it first obtains a no-context draft, extracts semantic signals from a multilingual embedding model (Zhang et al., 2025) as well as structured signals (lengths, context size, language leak checks, estimated cost), and then selects the contextual intervention for finally generating only that output. Note that draft call is included in the cost. We report standard MT metrics BLEU (Papineni et al., 2002), chrF++ (Popović, 2017), and COMET (Rei et al., 2020) alongside the final cost separately from the routing objective.

We evaluate the router on a filtered context-aware subset of the following datasets: TED (Reimers and Gurevych, 2020), Open-Subtitles (Lison and Tiedemann, 2016), and QED (Abdelali et al., 2014) covering En→Ja, En→De, De→En, and Zh→Ja. Figure 1 summarizes the main result across cost and performance. Fixed no-context translation is cheap but misses context-dependent gains while fixed context-aware and refinement policies improve some reference metrics but spend more tokens and can reduce deployment utility. Meanwhile, our CONTEXTROUTER improves both utility and reference metrics over these fixed policies while using fewer tokens than always context-aware translation. Moreover, CONTEXTROUTER-Q, a quality-oriented variant of the router, obtains stronger metric scores at higher cost, hinting at a quality–cost tradeoff rather than a single universal best policy. Our contributions are twofold:

1. We formulate LLM-MT as a test-time inter-

vention routing problem and introduce CONTEXTROUTER, a group-relative route policy that learns when each context intervention is worth applying using only test-time-available signals.

2. We show that routing improves the quality-cost tradeoff over fixed contextual policies. Specifically, a utility-oriented router improves both deployment utility and standard MT metrics, while a quality-oriented variant and oracle analysis reveal the cost tradeoff of pursuing higher reference-metric scores.

2 CONTEXTROUTER: Routing Translation Interventions

CONTEXTROUTER is an offline routing policy for deciding when an LLM-MT system should translate directly, use previous source-side context, or refine a context-aware MT draft. During training, we generate one candidate translation for each contextual intervention and score these candidates with a routing reward. At inference time, the router sees only test-time-available information comprising the current source segment, previous source-side context, target language, and a direct no-context draft. It then either returns this no-context draft or executes a single selected context intervention. The generator and embedding model remain frozen and only the route policy is trained.

2.1 Task and Action Space

For each example, let x be the current source segment, $C = (c_1, \dots, c_k)$ the preceding source-side context, ℓ the target language, and r the reference translation available only during training and evaluation. The system must output one target-language translation y . We keep the translation generator G fixed and define a finite action space of translation interventions:

$$\mathcal{A} = \{\text{NO-CONTEXT}, \text{CONTEXT-AWARE}, \text{CONTEXT-REFINE}\}. \quad (1)$$

Each action $a \in \mathcal{A}$ corresponds to a prompting-and-generation procedure G_a . NO-CONTEXT translates x directly. CONTEXT-AWARE translates x with preceding source-side context C . CONTEXT-REFINE first obtains a context-aware draft and then asks the same generator to refine it using x , C , and the draft. During training and validation, we generate all action candidates,

$$y_a = G_a(x, C, \ell),$$

so that route supervision can compare interventions for the same example. At test-time, however, the router does not generate all candidates, it first obtains a direct NO-CONTEXT draft and then executes only the selected route intervention.

2.2 Routing Objective

We train the router with a routing objective, not a new MT metric. The objective assigns each candidate contextual intervention a reward that combines three crucial quantities: translation quality, token cost, and failure checks, following the broader translation-evaluation practice of assessing task-relevant error types rather than relying only on a single aggregate score (Lommel et al., 2013; Guerreiro et al., 2024). This allows the router to learn the operating tradeoff faced at test time of when an action should improve translation quality enough to justify its additional cost.

Quality. For a candidate y_a with reference r , we define a reference-aware quality score

$$Q_\alpha(y_a, r) = \alpha_C \widehat{\text{COMET}}(x, y_a, r) + \alpha_F \widehat{\text{chrF++}}(y_a, r) + \alpha_B \widehat{\text{BLEU}}(y_a, r). \quad (2)$$

where metrics are normalized to comparable scales. We use COMET (Rei et al., 2020), chrF++ (Popović, 2017), and BLEU (Papineni et al., 2002) as standard reference-based MT metrics. COMET is used as the dominant semantic quality signal, while chrF++ and BLEU provide complementary lexical and surface-form evidence as prior work has shown that overlap-based metrics can remain sensitive to deviations even when neural metrics are stronger overall (Glushkova et al., 2023).

Failure Risks. Some MT outputs have severe problems that are not well summarized by average reference metrics. We therefore subtract a small penalty as risk for three heuristic validation checks:

$$R_\gamma(y_a, x, \ell) = \gamma_{\text{leak}} R_{\text{leak}}(y_a, x) + \gamma_{\text{len}} R_{\text{len}}(y_a, x) + \gamma_{\text{script}} R_{\text{script}}(y_a, \ell). \quad (3)$$

where R_{leak} flags source-language leakage, R_{len} flags severe length-ratio failures, and R_{script} flags target-script mismatch. These validators are heuristic safeguards rather than a replacement for MT evaluation with further details given in Appendix C.

Route cost. Let $T_{\text{exec}}(a)$ be the input + output tokens required to execute action a under its own prompt template and call sequence. For CONTEXT-REFINE, this includes both the initial context-aware draft and later refinement call. Learned router additionally pays for the no-context draft used to compute routing features. Thus, for a learned router selecting action a , the routed cost is

$$T_{\text{route}}(a) = T_{\text{exec}}(\text{NO-CONTEXT}) + \mathbb{I}[a \neq \text{NO-CONTEXT}] T_{\text{exec}}(a). \quad (4)$$

We normalize this cost with

$$K(a) = \min\left(\frac{T_{\text{route}}(a)}{B}, 1\right), \quad (5)$$

where B is a development-set operating budget. The cap prevents a small number of very long contexts from dominating route training, while raw tokens are still reported separately in all results. Fixed policies are charged for the action they actually execute, whereas learned routers and routing oracles are charged according to the route-only protocol in Section 2.5.

The overall utility reward used by CONTEXTROUTER is hence

$$U(y_a, a) = [Q_\alpha(y_a, r) - \lambda K(a) - \mu R_\gamma(y_a, x, \ell)]. \quad (6)$$

The weights α , γ , λ , μ and budget B are selected on validation data as reported in the Appendix C. We use U only for router training and model selection while also reporting the standard MT metrics and final tokens separately.

2.3 Routing Signals

The router is trained using rewards computed from references, but at test-time it can only use signals available before any context intervention is generated. For each test example, we first generate a no-context draft

$$y_{\text{nc}} = G_{\text{NO-CONTEXT}}(x, \ell).$$

The router then constructs semantic and structured routing signals from (x, C, y_{nc}) .

Semantic routing signals. We compute semantic signals with a frozen multilingual embedding model E , instantiated as Qwen3-Embedding-0.6B in our experiments (Zhang et al., 2025). We encode

the current source segment, each previous context turn, and the no-context draft:

$$e_x = E(x), \quad e_i = E(c_i), \quad e_{\text{nc}} = E(y_{\text{nc}}). \quad (7)$$

Following bi-encoder semantic-similarity methods (Feng et al., 2022; Reimers and Gurevych, 2019), we use cosine similarity to estimate source-context relevance

$$\rho_i = \cos(e_x, e_i). \quad (8)$$

We then aggregate previous context turns by cosine-similarity attention

$$\alpha_i = \frac{\exp(\rho_i/\tau_c)}{\sum_{j=1}^m \exp(\rho_j/\tau_c)}, \quad e_C = \sum_{i=1}^m \alpha_i e_i, \quad (9)$$

where τ_c is a context-pooling temperature tuned on development data. The semantic feature vector is

$$\phi_{\text{sem}}(s) = [e_x; e_C; e_{\text{nc}}; \max_i \rho_i; m^{-1} \sum_i \rho_i; \cos(e_x, e_{\text{nc}})]. \quad (10)$$

This gives our router the semantic content of the current source, a relevance-weighted representation of the previous context, as well as a coarse cross-lingual relation between the source and the no-context draft.

Structured routing signals. Structured features include non-embedding information available before the selected non-direct intervention is generated such as source length, context length, number of previous context turns, script ratios, lexical and named-entity overlap, estimated prompt tokens and call counts for each intervention, and no-context draft information such as output length, source-language leakage, severe length-ratio failure, and script mismatch. We exclude corpus/domain identifiers and generator identity features to reduce dataset- or model-specific shortcuts. We denote the structured action feature vector by $\phi_{\text{struct}}(s, a)$, it includes both segment-level signals as well as action-level estimates such as the token cost of intervention a .

2.4 Group-Relative Route Learning

CONTEXTROUTER is a routing policy over translation interventions. For state $s = (x, C, y_{\text{nc}}, \ell)$ and action a , we construct

$$z(s, a) = [\phi_{\text{sem}}(s); \phi_{\text{struct}}(s, a); \text{onehot}(a)]. \quad (11)$$

A small multilayer perceptron maps this vector to an action score for an intervention,

$$h_\theta(s, a) = f_\theta(z(s, a)). \quad (12)$$

and the route policy is the softmax over actions:

$$\pi_\theta(a | s) = \frac{\exp(h_\theta(s, a))}{\sum_{a' \in \mathcal{A}} \exp(h_\theta(s, a'))}. \quad (13)$$

In our main experiments, f_θ is a MLP with hidden sizes 512 and 128. We use a small MLP because routing depends on interactions among semantic context relevance, structured validators, and test-time cost (Ong et al., 2025; Shazeer et al., 2017). For example, relevant context may be useful when the direct draft is uncertain, but unnecessary when the draft is already clean and the context action is expensive. Further details are reported in Appendix D.

For each training example, we have one reward r_a for every action. For CONTEXTROUTER, $r_a = U(y_a, a)$, where U is the routing utility from Section 2.2. We also train CONTEXTROUTER-Q, a quality-oriented variant that uses a quality-heavy reward with targeted risk penalties but no default token-cost penalty.

Furthermore rather than training on only the single best action, we standardize rewards within each example’s candidate group:

$$A_a = \frac{r_a - \mu_r}{\sigma_r + \epsilon}, \quad \mu_r = \frac{1}{|\mathcal{A}|} \sum_{a' \in \mathcal{A}} r_{a'}. \quad (14)$$

Here σ_r is the within-example reward standard deviation and ϵ is a scale floor. The resulting advantage A_a asks which intervention is preferable for this source segment and context, with supervision strength that is comparable across language pairs and domains, while genuinely near-tied groups receive softened targets.

We convert these advantages into a soft route target:

$$q(a | s) = \frac{\exp(A_a/\tau)}{\sum_{a' \in \mathcal{A}} \exp(A_{a'}/\tau)}, \quad (15)$$

where τ controls target sharpness. Low τ approaches a hard winner-take-all label while higher τ preserves uncertainty when actions are all nearly tied.

The router is trained with cross-entropy to the group-relative target and a KL regularizer:

$$\begin{aligned} \mathcal{L} = & - \sum_{a \in \mathcal{A}} q(a | s) \log \pi_\theta(a | s) \\ & + \beta_{\text{KL}} \text{KL}(\pi_\theta(\cdot | s) \| \pi_{\text{ref}}(\cdot)). \end{aligned} \quad (16)$$

where π_{ref} is uniform over valid actions. With a uniform reference, the KL term discourages overconfident routing unless the group-relative target provides strong evidence for a particular contextual intervention.

The objective is listwise within each example as it asks which intervention is best for the current source segment and context, rather than predicting absolute reward on a global scale. This is useful for contextual intervention routing because context and refinement often yield small, instance-specific tradeoffs rather than universally superior outputs. Although this is related to preference optimization, and GRPO-style reward normalization (Ramesh et al., 2024; Rafailov et al., 2023), unlike previous work, we do not update the translation model. The generator and embedding model remain frozen and only the small route policy is trained.

2.5 Route-Only at test and Cost Accounting

At test-time, CONTEXTROUTER follows a route-only protocol. First, the system generates a no-context draft y_{nc} . Second, it extracts semantic and structured routing signals. Finally, it chooses the action for intervention

$$\hat{a} = \arg \max_{a \in \mathcal{A}} \pi_\theta(a | x, C, y_{\text{nc}}, \ell). \quad (17)$$

If $\hat{a} = \text{NO-CONTEXT}$, the no-context draft is returned as is. Otherwise, the system generates only the selected context intervention and returns that output.

For any generation procedure, the raw token cost is the sum of input and output tokens across all required model calls:

$$T(a) = \sum_{j \in \text{calls}(a)} (T_j^{\text{in}} + T_j^{\text{out}}). \quad (18)$$

For CONTEXTROUTER, test-time cost includes the initial no-context call used for routing:

$$\begin{aligned} T_{\text{deploy}}(\hat{a}) = & T_{\text{NO-CONTEXT}}^{\text{draft}} \\ & + \mathbb{I}[\hat{a} \neq \text{NO-CONTEXT}] T(\hat{a}). \end{aligned} \quad (19)$$

Calls at test-time are counted analogously. This accounting prevents the router from receiving a free no-context draft and prevents evaluation from assuming access to non-existing action outputs. Oracle policies, meanwhile, are evaluated separately as upper bounds for routing. They select the best action after all candidates have been generated and scored with references.

3 Experimental Setup

Our experiments test whether instance-level contextual intervention routing improves the quality–cost tradeoff for LLM-MT. We compare our learned CONTEXTROUTER against fixed intervention policies, a score-regression router, as well as upper-bound oracle policies. All systems use the same ordered source-side context, action definitions, prompt templates, decoding settings, and raw token accounting as discussed in Section 2.

3.1 Evaluation Data

We evaluate on a filtered ordered-context subset of publicly available datasets TED, QED, and OpenSubtitles. Filtering retains examples with recoverable previous source-side context and removes malformed instances. Appendix A covers the full construction procedure and corpus-by-direction split. Our final evaluation set contains 10,000 context-applicable examples across four directions: En→Ja, En→De, De→En, and Zh→Ja. Each example has previous source-side context from the same ordered unit, such as a talk, video, document, or subtitle group. TED provides cleaner talk/document context, QED provides educational subtitle context, while OpenSubtitles provides noisier real-life dialogue subtitle context. We cap the previous context window at $k = 5$ segments.

3.2 Models

Our main generator is Qwen3.6-27B (Qwen Team, 2026), served locally with vLLM (Kwon et al., 2023). We use Qwen3.6 for the full routing pipeline from generating the training candidates, scoring oracle candidates, training route policies, and evaluating the router at test-time. To test whether the learned routing behavior is tied to this generator, we additionally evaluate a Gemma4-31B-IT (Team et al., 2026) for transferability. The routing policy trained on Qwen candidates is transferred to Gemma4 using the same action intervention definitions and test-time constraints. Neither generator is fine-tuned.

All actions use fixed prompt templates and decoding settings. The main action space is the three-intervention set from Section 2.1: NO-CONTEXT, CONTEXT-AWARE, and CONTEXT-REFINE. Exact prompts and model decoding parameters are given in Appendix B.

3.3 Baselines, Oracles, and Learned Routers

Fixed policies. Fixed policies always choose one fixed contextual intervention: NO-CONTEXT, CONTEXT-AWARE, or CONTEXT-REFINE. They test whether no-context direct translation, always using context, or always refining is a good unconditional test-time routing strategy.

SCOREREG. SCOREREG tests whether routing can be solved by absolute utility prediction. For each action, it predicts that action’s deployment reward from the same available routing signals and selects the action with simply the highest predicted reward. Unlike CONTEXTROUTER, it does not train on within-example relative preferences among interventions.

QE-RERANK. QE-RERANK tests whether the routing problem can instead be solved by generating multiple candidates and selecting among them with a QE model. QE-RERANK-2 generates the NO-CONTEXT and CONTEXT-AWARE outputs and returns the candidate with the higher COMET-Kiwi (Rei et al., 2022) score. QE-RERANK-3 additionally generates CONTEXT-REFINE and selects among all three.

Oracles. Oracle policies are evaluated as an upper bound for our routing strategies. They select the best action after all candidate translations have been generated and scored with references. We report them only to measure remaining routing headroom.

CONTEXTROUTER. CONTEXTROUTER is our proposed router from Section 2, trained and selected by routing utility. We also train and compare another variant, CONTEXTROUTER-Q, which uses the same model and features but is selected with the quality-oriented reward described in Section 2. It is included as an analysis variant showing how routing changes when reference-metric gains are prioritized over token cost.

3.4 Evaluation Metrics

Learned routers are evaluated with the route-only protocol from Section 2. For each test example, the system first generates a no-context draft, extracts the routing signals, selects one intervention, and returns either the draft or the selected contextual intervention output. We report the utility reward from Section 2 as a routing objective, but not as a standalone MT metric. Each result also

reports standard reference-based MT metrics including COMET, chrF++, and BLEU as well as token cost. Fixed policies are charged for the calls they actually execute while CONTEXTROUTER is charged for the direct no-context draft plus any selected intervention.

4 Results and Analysis

We test whether our CONTEXTROUTER with its relevant contextual interventions improves the quality-cost tradeoff for LLM-MT over fixed and reranking baselines. Section 4.1 covers the main router evaluation results, Section 4.2 explains the quality-cost frontier and role of refinement, Section 4.3 evaluates transferability to other models, Section 4.4 studies the effect of various routing signals, and Section 4.5 ends with a small scale human evaluation that further verifies the router’s interventions.

4.1 Routing improves over fixed policy baselines in every language direction

Table 1 reports router evaluation on Qwen3.6-27B. Compared with NO-CONTEXT, CONTEXTROUTER raises utility score from 58.18 to 59.05, chrF++ from 35.02 to 37.18, and COMET from 80.80 to 81.68. Compared with fixed CONTEXT-AWARE translation, it improves all quality metrics while reducing test-time tokens from 112.0 to 85.9 per example. Thus, the router does not merely save tokens, it selects context and refinement as interventions when they are expected to improve the output enough to justify their cost.

The fixed policies show why unconditional context use is insufficient. CONTEXT-AWARE improves chrF++ and COMET over NO-CONTEXT, but lowers utility and more than doubles token cost. CONTEXT-REFINE improves some reference metrics further, but is substantially more expensive and has even lower utility. SCOREREG, which predicts each action’s absolute utility independently, gives only a small improvement over NO-CONTEXT suggesting that routing signals are useful, but that within-example relative supervision is a better fit for choosing among interventions.

Table 1 also compares against QE reranking. Both rerankers improve reference metrics over their underlying fixed policies, confirming that selection among candidates helps. However, generating every candidate before selecting is expensive as QE-RERANK-2 spends 158.1 tokens per example and QE-RERANK-3 spends 293.8. As a result, CON-

TEXTROUTER surpasses both rerankers on every metric, with QE-RERANK-2 trailing at 1.8x the tokens, and QE-RERANK-3 gaining only marginally over CONTEXT-REFINE at 3.4x cost. One plausible explanation is that segment-level QE does not observe the preceding context and therefore may not reward discourse-motivated deviations.

The language-pair breakdown shows that the aggregate result is not driven by a single direction. CONTEXTROUTER improves utility over both NO-CONTEXT and CONTEXT-AWARE for De→En, En→De, En→Ja, and Zh→Ja, and improves chrF++ in all four directions. The change is largest for the German directions, while Japanese-target directions remain harder because many subtitle references are short or compressed. The quality variant, CONTEXTROUTER-Q occupies a different operating point. It obtains the strongest test-time reference-metric scores in Table 1, but spends 142.4 tokens per example and has lower utility than the main router. This is the intended behavior as removing the default token-cost penalty makes the policy more willing to pay for context and refinement. We therefore treat CONTEXTROUTER as the default routing policy and CONTEXTROUTER-Q as a quality-seeking variation.

4.2 Routing reveals when context is worth using

Figure 1 visualizes the overall quality-cost frontier. Fixed NO-CONTEXT translation is cheap but misses context-dependent improvements, while unconditional CONTEXT-AWARE and CONTEXT-REFINE spend many more tokens. Our CONTEXTROUTER moves upward from NO-CONTEXT and remains left of fixed CONTEXT-AWARE. CONTEXTROUTER-Q moves further upward but also rightward, showing that stronger raw metrics require additional computation.

Action distributions in Figure 3 explain this curve where utility-oriented policies are conservative but not anti-context. Oracle-U, the upper bound for utility routing, selects NO-CONTEXT for 74% of examples, CONTEXT-AWARE for 24%, and CONTEXT-REFINE for 2%. CONTEXTROUTER closely tracks this pattern (72%, 25%, and 3%). Under the quality-oriented objective, both learned and oracle policies invoke interventions more often with CONTEXTROUTER-Q selecting CONTEXT-AWARE for 49% and CONTEXT-REFINE for 10% of examples suggesting that context and refinement are often useful, but their value is instance-

System	chrF++ $\uparrow$					All			
	All	De $\rightarrow$ En	En $\rightarrow$ De	En $\rightarrow$ Ja	Zh $\rightarrow$ Ja	Utility $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$	Tok. $\downarrow$
NO-CONTEXT	35.02	56.98	52.31	14.54	10.56	58.18	20.89	80.80	46.1
CONTEXT-AWARE	35.77	57.72	53.23	15.31	11.00	57.10	21.93	81.03	112.0
CONTEXT-REFINE	36.18	58.22	53.75	15.62	11.28	55.45	22.28	81.27	247.7
SCOREREG	35.42	57.37	52.80	14.96	10.80	58.19	21.38	80.95	60.5
QE-RERANK-2	35.94	58.33	53.63	15.08	10.86	56.53	21.72	81.30	158.1
QE-RERANK-3	36.20	58.70	54.00	15.23	10.95	55.82	21.95	81.45	293.8
CONTEXTROUTER	37.18	60.52	55.54	15.58	11.15	59.05	22.95	81.68	85.9
CONTEXTROUTER-Q	37.67	60.61	55.63	16.56	11.85	57.91	23.34	81.82	142.4
Oracle-U	37.27	60.45	55.48	15.72	11.35	59.35	23.39	81.87	80.8
Oracle-Q	<u>37.69</u>	<u>60.65</u>	<u>55.68</u>	<u>16.58</u>	<u>11.88</u>	58.76	<u>23.67</u>	81.62	119.8

Table 1: Quality-cost results on Qwen3.6-27B. chrF++ is shown overall and by language pair; utility, BLEU, COMET, and input-plus-output tokens are aggregated over all pairs. CONTEXTROUTER is the best utility-oriented routing policy, improving quality over fixed NO-CONTEXT and CONTEXT-AWARE while remaining cheaper than always using context. CONTEXTROUTER-Q is a quality-oriented variant that spends more tokens for stronger reference-metric scores. QE-RERANK selects among generated candidates by COMETKiwi, improving over the fixed policies it selects from but at substantially higher token cost. Oracle rows are upper bounds.

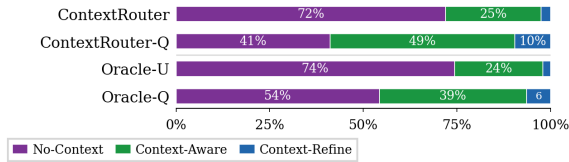


Figure 3: Action distributions on the Qwen test split. Utility-oriented policies are no-context-heavy, while quality-oriented policies invoke context and refinement more often.

dependent. Note that rarity of CONTEXT-REFINE reflects its sparse payoff rather than a redundant action space as refinement improves the context-aware output in only 7.7% of cases, but with a large conditional gain. A two-action ablation (Appendix K.2) confirms that refinement’s incremental value concentrates at quality-oriented operating points.

Qualitative examples in Table 2 reflects our router behavior. Context helps when it resolves discourse-dependent references, but it can also introduce unsupported lexical or gender shifts. Refinement can fix literal drafts, but the utility-oriented router invokes it rarely due to higher cost.

4.3 Routing policy transfers

Table 4 tests whether the learned routing behavior is specific to the generator model, Qwen3.6-27B. We apply the Qwen-trained route policy to Gemma4-31B without fine-tuning the generator and find the same qualitative pattern. The fixed CONTEXT-AWARE and CONTEXT-REFINE policies improve some reference metrics over NO-CONTEXT, but reduce test-time utility because of their token cost. Meanwhile, transferred CONTEXTROUTER improves over Gemma NO-CONTEXT on utility, chrF++, BLEU, and COMET, while us-

Phenom.	Example	Routing
Context helps	Zh $\rightarrow$ Ja: ctx : 在，我要你取个名字... src : 皮曹！ No-ctx transliterates as ピノ・ツアオ; ctx recovers the conventional Japanese ピノキオ (<i>Pinocchio</i>).	Both routers select CONTEXT-AWARE; discourse resolves the entity.
Context hurts	En $\rightarrow$ De: ctx : “Mr. Franzis... Mr. Alan is dead... murdered!” src : “The somnambulist’s prophecy...?” No-ctx: <i>des Somnambulen</i> (correct); ctx: <i>der Schlafwandlerin</i> (shifts gender without support).	CONTEXTROUTER selects NO-CONTEXT; CONTEXT avoids the unsupported shift.
Context-Refine helps	En $\rightarrow$ De: src : “That’s the thing.” Draft: <i>Das ist die Sache</i> (literal); refine: <i>Das ist der Punkt</i> (idiomatic).	CR routes to CONTEXT-REFINE sparingly (3% of examples).
U vs. Q trade-off	En $\rightarrow$ De: src : “Global Fund for Women in San Francisco.” No-ctx: <i>Globaler Fonds für Frauen</i> (translates name); ctx: preserves the English name, matching reference.	CR: NO-CONTEXT (cheaper); CR-Q: CONTEXT-AWARE (surface match).

Table 2: Representative routing examples from the test set. Each row illustrates a routing phenomenon: when context helps or hurts, when refinement is justified, and how utility- and quality-oriented routers diverge.

ing fewer tokens than fixed CONTEXT-AWARE setting. These results suggest that the router captures reusable intervention-selection behavior, although broader transfer tests are left as future work.

4.4 Structured and semantic signals are complementary

Table 3 ablates the two main routing-signal families. The structured-only router uses lengths, costs,

System	Utility $\uparrow$	chrF++ $\uparrow$	COMET $\uparrow$	Tok. $\downarrow$
Structured-only	58.42	35.72	81.08	62.3
Semantic-only	58.95	36.95	81.58	91.2
CONTEXTROUTER	59.05	37.18	81.68	85.9

Table 3: Feature ablation on Qwen3.6-27B. Structured-only routing is cheaper but misses quality gains; semantic-only routing improves quality but spends more tokens. The full CONTEXTROUTER combines semantic relevance with structured cost and draft-risk signals, yielding the best quality–cost tradeoff.

System	Utility $\uparrow$	chrF++ $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$
NO-CONTEXT	59.01	36.53	22.21	81.15
CONTEXT-AWARE	57.91	37.28	22.72	81.36
CONTEXT-REFINE	56.25	37.58	23.02	81.58
CONTEXTROUTER	60.06	38.05	23.18	81.82
CONTEXTROUTER-Q	59.12	38.91	24.27	82.42
Oracle-U	60.23	38.91	24.59	82.17
Oracle-Q	59.63	39.40	24.93	81.94

Table 4: Aggregate evaluation scores for cross-model transfer on Gemma4-31B. Routing policies applied without any generator fine-tuning. CONTEXTROUTER-Q again maximises reference-metric quality, exceeding the utility-oriented oracle on chrF++ and BLEU.

script features, and direct no-context draft. It keeps the token cost low, but misses many quality improvements. The semantic-only router uses embedding-based source–context and source–draft signals. It recovers more quality, but spends more tokens than the full router because semantic relevance alone does not regulate cost. CONTEXTROUTER combines both signal families and achieves the best score across utility, chrF++, and COMET. Appendix H further varies the token-cost penalty and cost-normalization budget.

4.5 Human evaluation of routing decisions

As the automatic reference metrics partially overlap with the routing reward, to further support the results we validate the routing decisions with a small human evaluation. Annotators judge the router’s chosen output against the no-context draft it rejected where the selected output is the context-aware translation for roughly 97% of routed items and the refined translation for the remainder, and annotators never see which action intervention produced it. The study covers En→De and En→Ja with three bilingual annotators per direction (six in total), ternary judgments with explicit ties, order-randomized and system-blind presentation with the preceding context displayed, and item labels by majority vote over 400 pairwise items (1,200 judgments). The routed cell is every test item where the router intervened, matching the 28% intervention

Direction	Cell	Agree / Disagree / Tie	Split κ
En→De	Routed ($n = 147$)	73 / 29 / 45	72% 0.57
En→Ja	Routed ($n = 133$)	54 / 25 / 54	68% 0.32
En→De	Declined ($n = 60$)	20 / 14 / 26	59% 0.39
En→Ja	Declined ($n = 60$)	14 / 18 / 28	44% 0.39

Table 5: Human evaluation of routing decisions. Annotators judge the router’s chosen output against the rejected alternative, blind to system identity. Agree means the majority preferred the router’s choice. Decisive split excludes ties.

rate from Section 4.2, and the declined cell samples items where the router returned the draft, judged against the context-aware candidate it declined.

Table 5 reports the results. Where the router selected an intervention, annotators prefer its choice in 72% (En→De) and 68% (En→Ja) of decisive judgments ($p=0.0015$). Where the router declined, we detect no significant preference for the contextual alternative ($p \geq 0.39$). For a cost-aware router, an intervention must beat the draft to justify its additional cost, so interventions are validated by preference and declines by the absence of a detectable benefit. Moreover the higher En→Ja tie rate is consistent with the subtitle compression discussed in Section 4.1. A complementary discourse analysis (Appendix J) shows the router intervenes twice more often on discourse-tagged segments.

5 Conclusion

We studied LLM-based MT as a test-time intervention routing problem. Rather than always translating directly, always appending context, or refining a draft, our CONTEXTROUTER learns which intervention to route for each source segment using only test-time signals. Across ordered-context examples from established MT corpora, fixed context and refinement policies improve some reference metrics but often incur enough token cost to reduce overall utility, and generating all candidates for QE reranking attains quality improvements only at substantially higher generation cost. In contrast, our group-relative router CONTEXTROUTER improves the quality–cost tradeoff by invoking context and refinement selectively, and human judgments validate these decisions as annotators prefer the router’s chosen output where it intervenes and show no detectable preference for the context it declines. The quality-oriented router and oracle analyses further show that raw metric improvements and utility favor different operating points. Overall, context and refinement are better treated as conditional test-time interventions for LLM-MT, not as default choices.

Limitations

This work studies when LLM-based MT systems should use context and self-refinement, but several limitations remain.

Our human evaluation is targeted rather than comprehensive. It validates the router’s escalate-versus-stop decisions on two directions (En→De and En→Ja) with 400 pairwise items and six annotators. Broader human studies, spanning more directions, professional evaluators, and error-typology annotation such as MQM, remain future work, and automatic reference metrics still cannot fully capture discourse adequacy, register, subtitle compression, or all acceptable translation variants.

The routing utility is necessarily a scalarization of several desiderata including reference-backed quality, applicable constraints, token cost, and targeted risks. Different deployment settings may prefer different tradeoffs. We address this partly through reporting individual metrics separately, but the chosen default utility still reflects one operating preference rather than a universal definition of translation quality.

Our evaluation covers a filtered subset of TED, QED, and OpenSubtitles across four translation directions using only preceding source-side context. This controlled setting may not generalize to technical, legal, medical, literary, or low-resource scenarios. Future work should explore richer document context, target-side history, longer windows, additional languages, and more diverse domains.

CONTEXTROUTER and all tested models may behave differently across domains or language pairs. The router can also make harmful errors such as selecting misleading context, skipping necessary context, or invoking unnecessary refinement. Although we evaluate transfer to a second generator, broader testing is needed before claiming model-agnostic routing.

Finally, our action space is limited to NO-CONTEXT, CONTEXT-AWARE, and CONTEXT-REFINE. Other interventions such as selected-context prompting, retrieval-augmented context, multi-draft reranking, or human-in-the-loop post-editing may be valuable and are left to future work.

Ethical Considerations

This work follows the ACL Ethics Policy and the ARR Responsible NLP Research checklist. The human evaluation in Section 4.5 involved six bilingual annotators (three per direction), as part of an

internal evaluation team made up of members with native or near native level target-language proficiency and they participated as part of their regular employment. Annotators judged machine translation outputs of publicly available corpus text and no personal or sensitive data was collected beyond language qualifications. They provided informed consent, and could withdraw at any time. The evaluation suite is constructed from publicly available OPUS-derived corpora, and any release of data, code, prompts, or model outputs should respect the licenses and redistribution conditions of the underlying corpora and models.

Machine translation systems can have real downstream consequences. Incorrect translations may mislead users, distort meaning, or cause harm in legal, medical, political, or safety-critical settings. Our method is intended as a research framework for studying context and refinement routing, not as a guarantee of safe or adequate translation. In particular, the router may still select an inappropriate action, preserve a misleading no-context draft, overuse context, or underuse context when discourse information is required.

The datasets used in this work include subtitles and public web-derived text, which may contain offensive language, stereotypes, fictional violence, sensitive topics, or biased portrayals. LLM generators may also reproduce or amplify social biases present in their training data. Routing does not remove these risks; it may change which risks appear by changing whether context or refinement is used. We therefore report validators and qualitative examples, but these should be understood as partial safeguards rather than comprehensive safety guarantees.

Finally, there are also resource and accessibility considerations. Generating multiple candidate translations for training and oracle analysis requires nontrivial compute, and two-stage actions increase inference cost when selected.

References

Ahmed Abdelali, Francisco Guzman, Hassan Sajjad, and Stephan Vogel. 2014. [The AMARA corpus: Building parallel language resources for the educational domain](#). In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC’14)*, pages 1856–1862, Reykjavik, Iceland. European Language Resources Association (ELRA).

Rachel Bawden, Rico Sennrich, Alexandra Birch, and

- Barry Haddow. 2018. [Evaluating discourse phenomena in neural machine translation](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1304–1313, New Orleans, Louisiana. Association for Computational Linguistics.
- Sheila Castilho and Rebecca Knowles. 2025. [A survey of context in neural machine translation and its evaluation](#). *Natural Language Processing*, 31(4):986–1016.
- Lingjiao Chen, Matei Zaharia, and James Zou. 2024a. [FrugalGPT: How to use large language models while reducing cost and improving performance](#). *Transactions on Machine Learning Research*. Featured Certification.
- Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2024b. [Iterative translation refinement with large language models](#). In *Proceedings of the 25th Annual Conference of the European Association for Machine Translation (Volume 1)*, pages 181–190, Sheffield, UK. European Association for Machine Translation (EAMT).
- Ritvik Choudhary, Rem Hida, Masaki Hamada, Hayato Futami, and Toshiyuki Sekiya. 2025. [Exploring context strategies in LLMs for discourse-aware machine translation](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 24382–24391, Suzhou, China. Association for Computational Linguistics.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. [Language-agnostic BERT sentence embedding](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 878–891, Dublin, Ireland. Association for Computational Linguistics.
- Zhaopeng Feng, Yan Zhang, Hao Li, Bei Wu, Jiayu Liao, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and ZuoZhu Liu. 2025. [TEaR: Improving LLM-based machine translation with systematic self-refinement](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 3922–3938, Albuquerque, New Mexico. Association for Computational Linguistics.
- Patrick Fernandes, Kayo Yin, Emmy Liu, André Martins, and Graham Neubig. 2023. [When does translation require context? a data-driven, multilingual exploration](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 606–626, Toronto, Canada. Association for Computational Linguistics.
- Baban Gain, Dibyanayan Bandyopadhyay, Asif Ekbal, and Trilok Nath Singh. 2026. [Bridging the linguistic divide: a survey on leveraging large language models for machine translation](#). *Language Resources and Evaluation*, 60(2).
- Javier Garcia Gilabert, Carlos Escolano, Xixian Liao, and Maite Melero. 2025. [Terminology-constrained translation from monolingual data using GRPO](#). In *Proceedings of the Tenth Conference on Machine Translation*, pages 1335–1343, Suzhou, China. Association for Computational Linguistics.
- Taisiya Glushkova, Chrysoula Zerva, and André F. T. Martins. 2023. [BLEU meets COMET: Combining lexical and neural metrics towards robust machine translation evaluation](#). In *Proceedings of the 24th Annual Conference of the European Association for Machine Translation*, pages 47–58, Tampere, Finland. European Association for Machine Translation.
- 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.
- Zhongzhan Huang, Guoming Ling, Yupei Lin, Yandong Chen, Shanshan Zhong, Hefeng Wu, and Liang Lin. 2025. [RouterEval: A comprehensive benchmark for routing LLMs to explore model-level scaling up in LLMs](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 3860–3887, Suzhou, China. Association for Computational Linguistics.
- Marzena Karpinska and Mohit Iyyer. 2023. [Large language models effectively leverage document-level context for literary translation, but critical errors persist](#). In *Proceedings of the Eighth Conference on Machine Translation*, pages 419–451, Singapore. Association for Computational Linguistics.
- Dayeon Ki and Marine Carpuat. 2024. [Guiding large language models to post-edit machine translation with error annotations](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4253–4273, Mexico City, Mexico. Association for Computational Linguistics.
- Sai Koneru, Miriam Exel, Matthias Huck, and Jan Niehues. 2024. [Contextual refinement of translations: Large language models for sentence and document-level post-editing](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2711–2725, Mexico City, Mexico. Association for Computational Linguistics.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with pagedattention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP ’23*, page 611–626, New York, NY, USA. Association for Computing Machinery.
- Pierre Lison and Jörg Tiedemann. 2016. [OpenSubtitles2016: Extracting large parallel corpora from](#)

- movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).
- Arle Richard Lommel, Aljoscha Burchardt, and Hans Uszkoreit. 2013. [Multidimensional quality metrics: a flexible system for assessing translation quality](#). In *Proceedings of Translating and the Computer 35*, London, UK. Aslib.
- Yingfeng Luo, Hongyu Liu, DingYang Lin, Kaiyan Chang, Chenglong Wang, Bei Li, Quan Du, Tong Xiao, and JingBo Zhu. 2026. [RouteLMT: Learned sample routing for hybrid LLM translation deployment](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 6: Industry Track)*, pages 1886–1897, San Diego, California, USA. Association for Computational Linguistics.
- Sameen Maruf, André F. T. Martins, and Gholamreza Haffari. 2019. [Selective attention for context-aware neural machine translation](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3092–3102, Minneapolis, Minnesota. Association for Computational Linguistics.
- Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. [RouteLLM: Learning to route LLMs from preference data](#). In *The Thirteenth International Conference on Learning Representations*.
- 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.
- Maja Popović. 2017. [chrF++: words helping character n-grams](#). In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.
- Qwen Team. 2026. [Qwen3.6-27B: Flagship-level coding in a 27b dense model](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Thirty-seventh Conference on Neural Information Processing Systems*.
- Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. 2024. Group robust preference optimization in reward-free rlhf. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- 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.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-BERT: Sentence embeddings using Siamese BERT-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.
- Nils Reimers and Iryna Gurevych. 2020. [Making monolingual sentence embeddings multilingual using knowledge distillation](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.
- Matiss Rikters and Toshiaki Nakazawa. 2024. [Revisiting context choices for context-aware machine translation](#). In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 14073–14079, Torino, Italia. ELRA and ICCL.
- 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.
- Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarczyk, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. [Outrageously large neural networks: The sparsely-gated mixture-of-experts layer](#). In *International Conference on Learning Representations*.
- Yanming Sun, Runzhe Zhan, Chi Seng Cheang, Han Wu, Xuebo Liu, Yuyao Niu, Fengying Ye, Kaixin Lan, Lidia S Chao, and Derek F Wong. 2026. Exposing the cracks: Vulnerabilities of retrieval-augmented llm-based machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33135–33143.

Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, et al. 2026. [Gemma 4 technical report](#). *Preprint*, arXiv:2607.02770.

David Vilar, Markus Freitag, Colin Cherry, Jiaming Luo, Viresh Ratnakar, and George Foster. 2023. [Prompting PaLM for translation: Assessing strategies and performance](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15406–15427, Toronto, Canada. Association for Computational Linguistics.

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

Longyue Wang, Chenyang Lyu, Tianbo Ji, Zhirui Zhang, Dian Yu, Shuming Shi, and Zhaopeng Tu. 2023. [Document-level machine translation with large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 16646–16661, Singapore. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. [Trans-formers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Di Wu, Seth Aycok, and Christof Monz. 2025. [Please translate again: Two simple experiments on whether human-like reasoning helps translation](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 20424–20440, Suzhou, China. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: a case study. In *Proceedings of the 40th International Conference on Machine Learning, ICML’23*. JMLR.org.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. [Qwen3 embedding: Advancing text embedding and reranking through foundation models](#). *Preprint*, arXiv:2506.05176.

Direction	Train	Dev	Test	Total
De→En	1539	512	449	2500
En→De	1457	518	525	2500
En→Ja	1502	523	475	2500
Zh→Ja	1691	433	376	2500
All	6189	1986	1825	10000

Table 6: Train/development/test split by translation direction. Splits are grouped by ordered unit.

Direction	Corpus	Train	Dev	Test	Total
De→En	TED	538	183	179	900
De→En	QED	540	183	177	900
De→En	OpenSubtitles	461	146	93	700
En→De	TED	538	183	179	900
En→De	QED	540	183	177	900
En→De	OpenSubtitles	379	152	169	700
En→Ja	TED	533	185	182	900
En→Ja	QED	539	180	181	900
En→Ja	OpenSubtitles	430	158	112	700
Zh→Ja	TED	701	112	87	900
Zh→Ja	QED	540	181	179	900
Zh→Ja	OpenSubtitles	450	140	110	700

Table 7: Corpus-by-direction composition. All retained examples contain previous source-side context.

A Dataset Construction and Splits

We evaluate on a filtered ordered-context subset of TED, QED, and OpenSubtitles. The subset is used as a controlled setting for comparing translation interventions when previous source-side context is available; it is not intended as a new general-purpose MT benchmark. All examples are context-applicable, meaning that at least one preceding source segment from the same ordered unit can be recovered. Ordered units correspond to talks, videos, documents, or subtitle groups, depending on the corpus.

Prompts include the current source segment and preceding source-side context only. They never include target-side context or reference translations. We cap the previous context window at $k = 5$ source-side segments. Train, development, and test splits are grouped by ordered unit so that adjacent segments do not cross splits. For $\text{En} \leftrightarrow \text{De}$, we additionally group reverse directions to avoid placing the same aligned segment in training for one direction and testing for the reverse direction.

The split tables are intended to make clear that the evaluation subset is balanced by translation direction but still drawn from heterogeneous context conditions. TED examples typically provide cleaner talk-level context, QED provides educa-

Step	Rule
Ordering recovery	Recover ordered units from corpus metadata: talks, videos, documents, or subtitle groups.
Context applicability	Retain examples with at least one previous source-side segment from the same ordered unit.
Context window	Use up to $k = 5$ previous source-side segments.
Malformed examples	Remove empty, missing, or alignment-corrupted examples.
Split grouping	Group by ordered unit; additionally group reverse En↔De aligned units.

Table 8: Filtering and split construction for the dataset.

tional subtitle context, and OpenSubtitles provides noisier dialogue-style subtitle context. This heterogeneity is useful for routing: a router should not assume that additional context is always equally relevant or equally safe across domains. Table 8 summarizes the preprocessing logic used to construct the subset.

The filtering rules are deliberately conservative. The goal is not to maximize corpus size, but to ensure that each retained example supports a controlled comparison between no-context translation, context-aware translation, and context-aware refinement. In particular, grouping by ordered unit prevents adjacent context windows from being split across train, development, and test partitions.

B Prompts and Generation Settings

All systems use the same prompt template for a given intervention. Fixed policies, learned routers, and oracle candidates differ only in which intervention output is selected. The prompts below show the intended structure.

NO-CONTEXT.

Instruction: Translate the following source segment into <TARGET_LANGUAGE>. Return only the translation.
Source: <SOURCE_SEGMENT>

CONTEXT-AWARE.

Instruction: Translate the current source segment into <TARGET_LANGUAGE> using the previous source-side context only when it is helpful. Return only the translation.
Previous context:
 <C_1>
 ...
 <C_k>
Current source: <SOURCE_SEGMENT>

CONTEXT-REFINE.

Instruction: Refine the draft translation into <TARGET_LANGUAGE> using the source segment and previous source-side context. Preserve meaning and return only the final translation.

Previous context:

<C_1>

...

<C_k>

Current source: <SOURCE_SEGMENT>

Draft translation: <CONTEXT_AWARE_DRAFT>

The three templates differ only in the information made available to the same frozen generator. This design isolates the intervention decision such that improvements from CONTEXTROUTER should come from choosing among interventions, not from using different prompt styles across systems. For CONTEXT-REFINE, the draft is the context-aware draft produced by the first stage of the same intervention. The No-Context draft used by the router is separate and is counted in deployment cost when learned routers are evaluated. We used the HuggingFace Transformers library (Wolf et al., 2020) along with vllm (Kwon et al., 2023) to run the models locally. We ran the model inference on two NVIDIA A6000 GPUs. The maximum output length was set to 1024 tokens, using the default parameters provided.

All fixed policies, learned routers, and oracle policies are evaluated with outputs produced under these same generation settings. Oracles differ only in selection: they select after all candidates have been generated and reference-scored, and are therefore non-deployable upper bounds rather than inference-time systems.

C Routing Objective, Metrics, and Validators

The objective is used to construct route supervision and to select checkpoints on development data. It is not a new general-purpose MT metric. The main tables therefore report utility, standard reference metrics, and token cost separately.

The coefficients for the quality, cost, and targeted risk term are selected on the validation set. The details are summarized in Table 9.

The capped token-cost penalty is

$$K(a) = \min\left(\frac{T_{route}(a)}{200}, 1\right), \quad (20)$$

where $T_{route}(a)$ is the input-plus-output token cost of action a under its prompt template and call sequence. Combining the above, the reported utility

Component	Value	Notes
COMET weight	0.45	Dominant semantic metric component.
chrF++ weight	0.25	Character/word surface-form component.
BLEU weight	0.10	N-gram surface-overlap component.
Cost penalty λ	0.05	Default deployment operating point.
Risk penalty scale μ	0.10	Applied to weighted validator risk R .
Cost denominator B	200	Token-cost cap denominator.
Leakage weight γ_{leak}	0.70	High-precision source-language leakage in the target flag.
Length weight γ_{length}	0.20	Severe length-ratio failure flag between the source and target.
Script weight γ_{script}	0.10	Target and source script mismatch flag.

Table 9: Routing utility settings used for route supervision and checkpoint selection.

is

$$U_{\text{report}}(y_a, a) = [Q(y_a, r) - \lambda K(a) - \mu R(y_a, x, \ell)] \quad (21)$$

A factor of 100 is used for table readability.

This objective should be read as an operating point. It encodes the preference that a deployable router should improve translation quality while avoiding unnecessary token cost and severe output failures. It is not intended to rank arbitrary MT systems outside this setting. For that reason, the main paper and appendix always report standard metrics and tokens separately from utility.

The risk validators (R) are intentionally narrow. They are used to discourage catastrophic outputs such as untranslated source fragments or wrong-script output, but they do not attempt to model all adequacy, fluency, or discourse errors.

D Router Features and Training Details

This section expands the routing-signal description from Section 2 of the main paper. The key constraint is the use of features at test. The router may use the current source segment, previous source-side context, the no-context draft, and quantities derived from them, but it may not use references, metric scores, oracle labels, or outputs from actions that have not been generated. The feature-availability distinction is central to the evaluation protocol. Candidate outputs for all actions are gen-

Component	Setting
Embedding model	Qwen3-Embedding-0.6B, frozen
Route scorer	MLP over action features
Hidden sizes	512, 128
Learning rate	3e-4
ϵ	0.5
τ_c	selected from {0.05, 0.1, 0.2, 0.5}
τ	selected from {0.5, 0.75, 1.0, 1.5}
β_{KL}	selected from {0, 0.01, 0.03, 0.05, 0.10}

Table 10: Router architecture and optimization settings.

erated during training and oracle analysis, but at inference time a learned router only sees test-time signals. This prevents the route policy from becoming a reranker over already generated translations.

The semantic features are meant to estimate whether the available context is relevant to the current segment and whether the no-context draft appears semantically compatible with the source. The structured features capture complementary evidence: length, script, No-Context-draft diagnostics, and estimated action cost. We exclude corpus, domain, and generator identifiers so that the policy is encouraged to learn from instance-level evidence rather than memorizing dataset or model priors.

The MLP size is kept small relative to the frozen generator and embedding model. Its role is to model interactions among semantic relevance, No-Context-draft diagnostics, and cost estimates, not to introduce a high-capacity translation model. Hyperparameters are selected on the development split: CONTEXTROUTER by development utility and CONTEXTROUTER-Q by the quality-oriented development reward.

E Full Qwen Results

Table 11 provides the Qwen3.6-27B route-only results underlying the compact main-paper presentation. To emphasize the quantities used in the analysis, we report utility, standard reference metrics, COMETKiwi (Unbabel/wmt22-cometkiwi-da), and deployed tokens. Action distributions are visualized separately in Appendix F. Throughout this appendix Tok. denotes deployed input-plus-output tokens per example.

The aggregate appendix table confirms the main-paper pattern. Under the default deployment utility, CONTEXTROUTER is the best deployable policy: it improves utility over the fixed policies and SCOREREG, while using fewer tokens than fixed CONTEXT-AWARE translation. In contrast, CONTEXTROUTER-Q pushes chrF++,

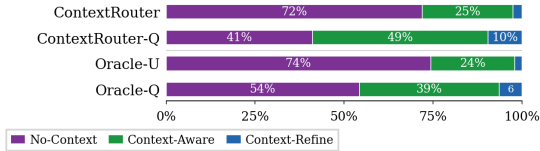


Figure 4: Stacked action distributions on the Qwen test split. Utility-oriented policies, CONTEXTROUTER, and Oracle-U, are conservative and choose NO-CONTEXT most often. Quality-oriented policies, CONTEXTROUTER-Q and Oracle-Q, invoke CONTEXT-AWARE and CONTEXT-REFINE more frequently. Counts are shown inside large segments.

BLEU, and COMET higher, but it does so at substantially higher token cost and lower utility. This supports the interpretation that CONTEXTROUTER is the default deployment policy, whereas CONTEXTROUTER-Q is a quality-seeking frontier point.

The language-pair table shows that the aggregate gains are not caused by a single direction. CONTEXTROUTER improves utility over both NO-CONTEXT and CONTEXT-AWARE for all four directions. The metric gains are largest for the German directions, while the Japanese-target directions remain more sensitive to reference omissions and shorter subtitle-style segments. The COMETKiwi column is included as an additional reference-free diagnostic; its behavior is not always identical to reference-based COMET, which reinforces the main paper’s decision to report multiple quality views rather than a single score.

F Oracle Headroom and Action Distributions

Action distributions make the quality–cost tradeoff more interpretable. A router that improves metrics by invoking context on nearly every example is different from one that uses context only when its expected gain justifies its cost. Figure 4 visualizes the action distributions of the learned policies and oracles on the Qwen test split.

The utility-oriented policies are conservative as both CONTEXTROUTER and Oracle-U choose NO-CONTEXT for most examples, while still using context or refinement for a substantial minority. In contrast, CONTEXTROUTER-Q and Oracle-Q invoke contextual interventions much more often. This explains why CONTEXTROUTER-Q obtains stronger reference metrics but pays more tokens and has lower deployment utility.

Figure 5 breaks the oracle distributions down by

language pair and domain group. This helps show where routing headroom comes from: the utility oracle remains No-Context-heavy even when context is available, while the quality-oriented oracle uses context and refinement more aggressively.

The subgroup breakdown shows the same qualitative pattern throughout the test set. Oracle-U chooses NO-CONTEXT most often in every language pair and domain group, indicating that context is useful but not universally worth its cost. Oracle-Q is consistently more intervention-heavy, especially in the Japanese-target and talk-style groups, where quality-oriented scoring finds more cases in which extra context or refinement can improve reference metrics.

G Gemma Transfer Results

Table 12 provides the full aggregate transfer results on Gemma4-31B in the same compact format as the Qwen appendix tables.

The transfer results repeat the Qwen pattern. Fixed context and refinement improve some reference metrics but reduce deployment utility because of their cost. Transferred CONTEXTROUTER improves utility and reference metrics over the Gemma NO-CONTEXT baseline while remaining cheaper than fixed CONTEXT-AWARE. CONTEXTROUTER-Q again behaves as a quality-seeking operating point: it spends more tokens to obtain stronger raw metrics.

The consistent direction of the aggregate result suggests reusable intervention-selection behavior, while broader transfer tests across additional generators remain future work.

H Sensitivity Analysis

The utility is an operating point rather than a universal MT quality metric. We therefore vary the token-cost weight λ_{cost} , the validator-risk scale μ , and the cost-normalization denominator d in Equation 5 to test whether the main qualitative conclusions depend on one narrow scalarization.

Figure 6 shows that the overall ranking of methods is stable, but the margin between them depends predictably on the deployment objective. As λ_{cost} increases, the utility of context-heavy fixed policies drops sharply, especially for CONTEXT-REFINE, while NO-CONTEXT becomes more competitive. Increasing the denominator d softens this effect by reducing the penalty assigned to expensive actions. Across the grid, CONTEXTROUTER remains close

LP	System	Utility $\uparrow$	chrF++ $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$	COMETKiwi $\uparrow$	Tok. $\downarrow$
De $\rightarrow$ En	NO-CONTEXT	70.21	56.98	34.62	86.67	83.50	55.26
De $\rightarrow$ En	CONTEXT-AWARE	68.96	57.72	35.85	86.90	83.50	128.1
De $\rightarrow$ En	CONTEXTROUTER	71.13	60.52	38.45	87.98	83.38	84.88
De $\rightarrow$ En	CONTEXTROUTER-Q	69.91	60.61	39.07	86.87	83.67	115.7
En $\rightarrow$ De	NO-CONTEXT	66.86	52.31	28.84	84.94	84.03	53.61
En $\rightarrow$ De	CONTEXT-AWARE	65.64	53.23	30.37	85.04	83.68	126.8
En $\rightarrow$ De	CONTEXTROUTER	68.08	55.54	31.99	86.28	83.94	85.01
En $\rightarrow$ De	CONTEXTROUTER-Q	66.47	55.63	32.54	86.64	83.86	116.6
En $\rightarrow$ Ja	NO-CONTEXT	48.83	14.54	10.49	78.07	85.60	41.31
En $\rightarrow$ Ja	CONTEXT-AWARE	47.73	15.31	11.17	78.43	85.18	110.8
En $\rightarrow$ Ja	CONTEXTROUTER	49.50	15.58	10.95	78.28	85.49	67.46
En $\rightarrow$ Ja	CONTEXTROUTER-Q	48.61	16.56	11.16	79.63	85.34	99.12
Zh $\rightarrow$ Ja	NO-CONTEXT	43.54	10.56	6.55	71.45	76.92	30.76
Zh $\rightarrow$ Ja	CONTEXT-AWARE	42.83	11.00	7.11	71.73	76.51	73.74
Zh $\rightarrow$ Ja	CONTEXTROUTER	44.08	11.15	6.87	71.63	76.88	48.35
Zh $\rightarrow$ Ja	CONTEXTROUTER-Q	43.36	11.85	7.11	71.83	76.58	69.12

Table 11: Qwen3.6-27B route-only results by language pair.

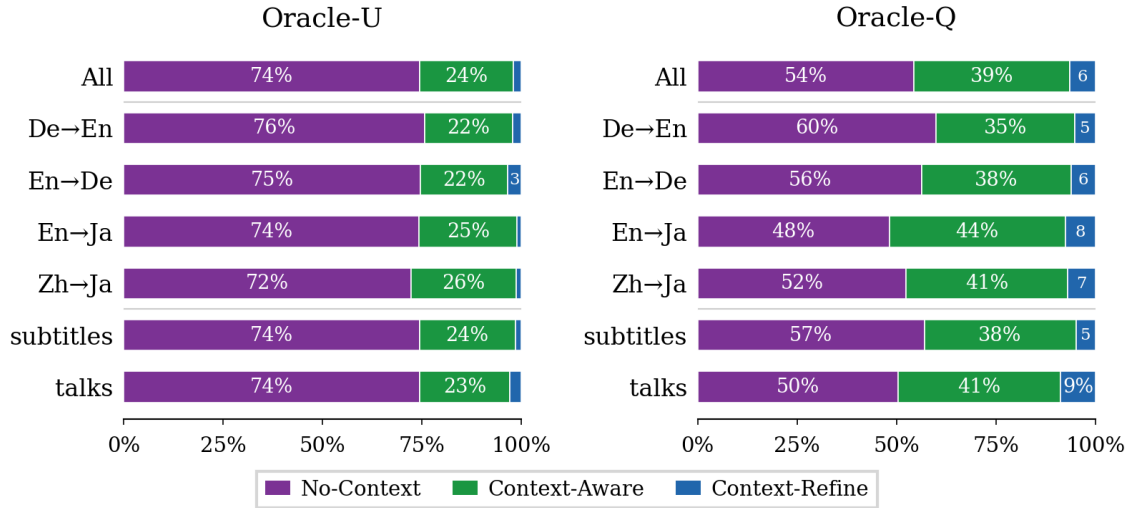


Figure 5: Oracle action distributions by language pair and domain group. Oracle-U remains conservative, while Oracle-Q increases both CONTEXT-AWARE and CONTEXT-REFINE usage across groups. Subtitles denote Open-Subtitles and QED-like subtitle conditions; talks denote TED-like talk conditions.

to the oracle and consistently outperforms the fixed deployable policies.

The effect of the validator-risk scale μ is comparatively modest. This is expected because severe validator-triggering failures are rare in the current test set, so most of the deployment tradeoff is driven by reference-backed quality and token cost. The highlighted panel corresponds to the paper setting. At this operating point, CONTEXTROUTER achieves the highest deployable utility, while quality-oriented routing would move toward a different point on the frontier.

Overall, the sensitivity analysis supports interpreting utility as a tunable deployment preference rather than a universal metric. When cost is penalized more strongly, No-Context-heavy policies

become preferable; when cost is penalized less strongly, context-aware actions become more attractive. The central qualitative conclusion remains unchanged: context and refinement are useful, but they should be invoked selectively.

I Human Evaluation details

This section documents the human evaluation reported in Section 4.5.

Samples Each sample is a pair consisting of the output CONTEXTROUTER returned and the alternative it rejected. For the routed cells we take a census of every test item where the router selected a contextual intervention (En $\rightarrow$ De n = 147 of 525; En $\rightarrow$ Ja n = 133 of 475), pairing the router’s out-

System	Utility $\uparrow$	chrF++ $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$	COMETKiwi $\uparrow$	Tok. $\downarrow$
NO-CONTEXT	59.01	36.53	22.21	81.15	82.71	46.19
CONTEXT-AWARE	57.91	37.28	22.72	81.36	82.62	111.6
CONTEXT-REFINE	56.25	37.58	23.02	81.58	82.72	246.5
Transferred CONTEXTROUTER	60.06	38.05	23.18	81.82	82.68	77.95
Transferred CONTEXTROUTER-Q	59.12	38.91	24.27	82.42	82.62	165.1
Oracle-U	60.23	38.91	24.59	82.17	82.86	81.82
Oracle-Q	59.63	39.40	24.93	81.94	82.75	120.1

Table 12: Aggregate Gemma4-31B transfer results.

LP	System	Utility $\uparrow$	chrF++ $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$	COMETKiwi $\uparrow$	Tok. $\downarrow$
De→En	Transferred CONTEXTROUTER	71.44	58.87	36.45	86.95	83.51	83.14
De→En	Transferred CONTEXTROUTER-Q	70.26	59.99	38.56	87.99	83.47	172.8
En→De	Transferred CONTEXTROUTER	68.89	55.65	31.95	85.55	83.89	80.49
En→De	Transferred CONTEXTROUTER-Q	67.38	56.26	33.01	86.47	83.95	168.3
En→Ja	Transferred CONTEXTROUTER	49.86	16.53	11.36	78.98	85.36	61.98
En→Ja	Transferred CONTEXTROUTER-Q	48.95	17.48	12.11	79.30	85.18	149.1
Zh→Ja	Transferred CONTEXTROUTER	44.72	12.52	8.02	71.78	76.63	46.11
Zh→Ja	Transferred CONTEXTROUTER-Q	43.74	13.23	8.64	72.06	76.53	99.24

Table 13: Gemma4-31B transfer results by language pair for the learned routers.

put with the no-context draft it rejected. For the declined cells we sample 60 items per direction where the router returned the no-context draft, pairing it with the context-aware candidate the router declined. This candidate was already generated during training and oracle scoring, so no additional generation was required.

Annotators Six annotators from an internal evaluation team participated, three per direction, with native or near-native proficiency in the target language and reading competence in the source language. A pilot of 10 items preceded the main task so that annotators could raise questions about the interface and the instructions.

Interface For each item, annotators saw the preceding source-side context (up to $k = 5$ segments, the same context available to the router), the current source segment, and the two candidate translations labeled only as A and B. System identity was hidden and the left-right order was randomized per item, so annotators could not infer which output came from which intervention. Judgments were ternary, with an explicit tie option, and annotators were clearly told that ties were an acceptable and expected outcome. Context was displayed on each turn.

Instructions Annotators received the following instructions:

You will see a short passage of preceding

context, one current sentence to be translated, and two candidate translations of that current sentence, labeled A and B.

Judge only the translation of the current sentence. The context is provided to help you decide whether a translation is appropriate in this discourse; do not judge how the context itself was translated.

Choose the translation you would prefer to receive as a reader of the target language, considering accuracy of meaning, appropriateness in the given context (for example pronouns, named entities, terminology, politeness level), and fluency. If the two translations differ but neither is clearly better, select Tie. Ties are a valid answer.

The two candidates were produced by automatic systems. Their order is randomized, and neither position indicates a preferred system.

Options: A is better / B is better / Tie

J Discourse Phenomenon Analysis

To test whether interventions align with discourse phenomena, we also tag the En→Ja test set ($n = 475$) with MuDA (Fernandes et al., 2023), whose taggers for this pair cover pronouns, lexical cohesion, and honorific formality (Table 14). The router intervenes 2.2x more often on tagged than untagged

MuDA bucket	n	CR intervenes	ΔchrF (CtxAware–No-CTx)	Tag F_1 , No-Context→Context-Aware→CONTEXTROUTER
pronoun	62	50%	+2.3	48.7→58.3→55.4
lexical cohesion	71	46%	+1.9	61.5→66.8→65.0
formality	181	34%	+1.1	69.8→73.4→72.2
any tag	247	38%	+1.4	—
no tag	228	17%	+0.1	—

Table 14: MuDA discourse analysis on En→Ja. The router intervenes far more often on discourse-tagged segments, and per-phenomenon intervention rates track the measured context gain.

System	Utility $\uparrow$	chrF++ $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$	Tok. $\downarrow$	Action mix
Structured-only	58.42	35.72	21.52	81.08	62.30	No-Context:1545, ctx:258, refine:22
Semantic-only	58.95	36.95	22.72	81.58	91.20	No-Context:1260, ctx:510, refine:55
CONTEXTROUTER	59.05	37.18	22.95	81.68	85.90	No-Context:1314, ctx:465, refine:46

Table 15: Ablation results on Qwen3.6-27B.

segments and the utility oracle shows the same improvement (about 36% vs 15%), indicating that the router recovers structure rather than imposing a preference. The per-phenomenon intervention rate shows the improvement as pronoun > cohesion > formality, with formality weakest, consistent with honorific register being largely resolvable within the segment.

K Ablations

K.1 Routing Signals

Table 15 studies the contribution of the two main routing signal families. The structured-only variant retains cost, length, script, and No-Context-draft diagnostics but removes the embedding-based semantic signals. The semantic-only variant retains the embedding-based routing signals but removes the structured features that make action cost and No-Context-draft diagnostics explicit. The full CONTEXTROUTER combines both.

The ablation results support a complementary view of the two feature families. The structured-only router is conservative: it chooses NO-CONTEXT for most examples and keeps token cost low, but misses many quality gains. The semantic-only router recovers much more of the quality improvement, indicating that embedding-based signals are important for identifying when previous context is relevant. However, it also uses more tokens than the full router, suggesting that semantic relevance alone is not sufficient to regulate cost effectively.

The full CONTEXTROUTER achieves the best utility and the strongest overall quality–cost trade-off. Relative to the semantic-only variant, it improves both utility and quality while using fewer

tokens. Relative to the structured-only variant, it uses more context and refinement, but does so selectively enough to recover substantially stronger chrF++, BLEU, and COMET. This is consistent with the design hypothesis in the main paper: semantic signals identify when context is useful, while structured signals help determine when that potential gain is worth the deployment cost.

K.2 Two-action ablation and refinement increment

We retrain CONTEXTROUTER and CONTEXTROUTER-Q with the action space restricted to NO-CONTEXT, CONTEXT-AWARE (5 seeds), and additionally evaluate a masked variant of the three-action router that never selects CONTEXT-REFINE. Table 16 reports results.

Per-example, the refinement increment $\Delta_{ref} = \text{quality}(\text{REFINE}) - \text{quality}(\text{CONTEXT-AWARE})$ is positive in 7.7% of cases with conditional mean about +4.7; the remaining mass is small, near one-sided degradation. Among CONTEXTROUTER-Q’s selected refine cases, 44% have a positive increment, with conditional mean +2.31 against +0.21.

System	Utility $\uparrow$	chrF++ $\uparrow$	BLEU $\uparrow$	COMET $\uparrow$	Tok. $\downarrow$
CR 3-action	59.05	37.18	22.95	81.68	85.9
CR 2-action (retrain)	59.06	37.14	22.88	81.65	81.3
CR 2-action (masked)	59.03	37.11	22.85	81.63	80.7
CR-Q 3-action	57.91	37.67	23.34	81.82	142.4
CR-Q 2-action (retrain)	58.07	37.51	23.12	81.74	126.1
CR-Q 2-action (masked)	58.02	37.47	23.06	81.72	124.8
Oracle-U 2-action	59.33	37.22	23.32	81.84	78.1
Oracle-Q 2-action	58.85	37.54	23.45	81.55	108.2

Table 16: Comparison of 3-action and 2-action routing variants.

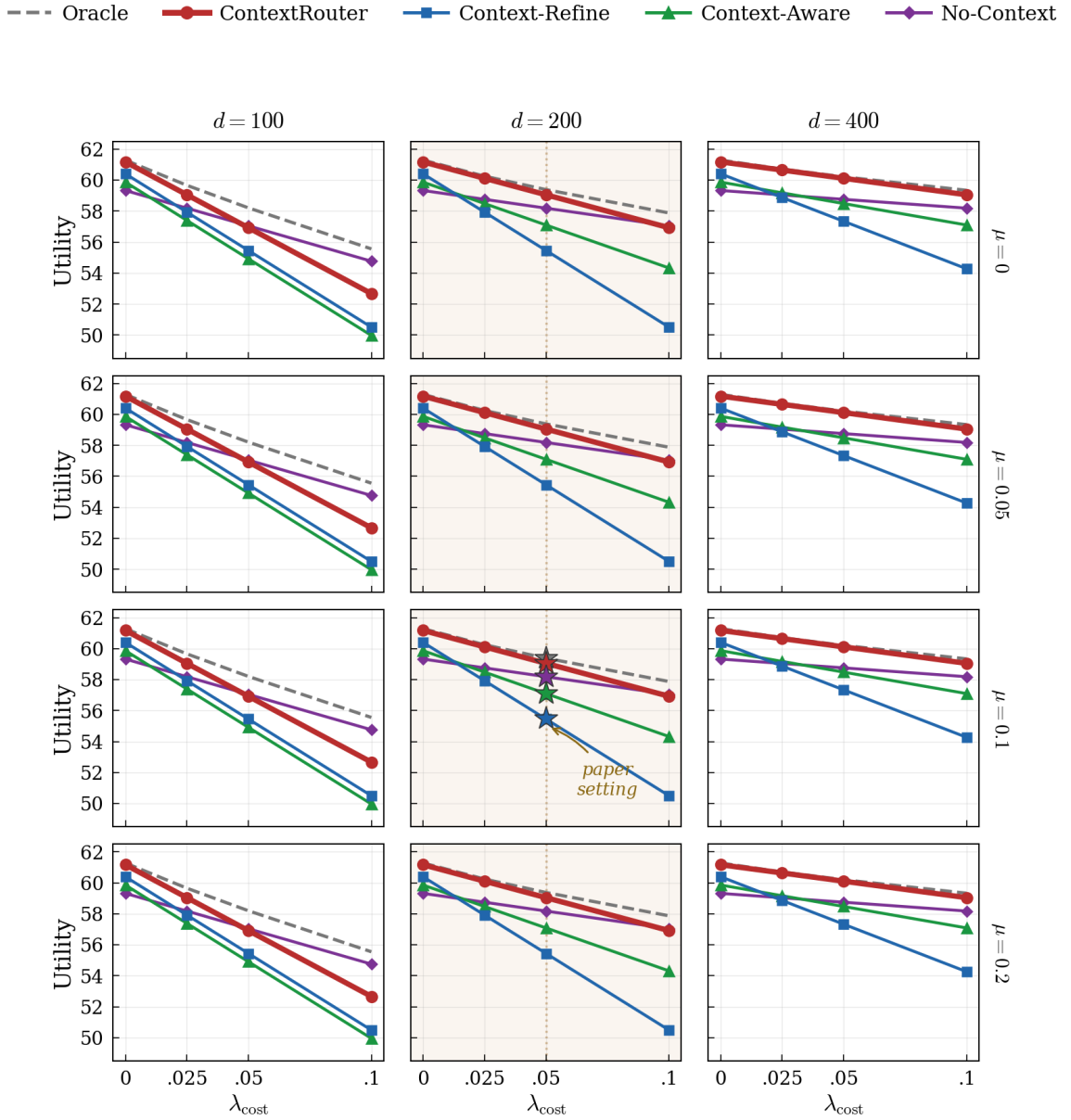


Figure 6: Utility under different deployment-objective settings. Columns vary the token-cost normalization denominator $d \in \{100, 200, 400\}$; rows vary the risk scale $\mu \in \{0, 0.05, 0.1, 0.2\}$; the x-axis varies the token-cost weight λ_{cost} . The shaded panel and starred markers denote the paper setting: $d = 200$, $\mu = 0.1$, and $\lambda_{\text{cost}} = 0.05$.