Early-Exit and Instant Confidence Translation Quality Estimation

Vilém Zouhar[1] **Maike Züfle**[2] **Beni Egressy**[3]
Julius Cheng[4] **Mrinmaya Sachan**[1] **Jan Niehues**[2]

[1]ETH Zurich [2]Karlsruhe Institute of Technology
[3]Heidelberg Institute for Theoretical Studies [4]University of Cambridge

Abstract

Quality estimation is omnipresent in machine translation, for both evaluation and generation. Unfortunately, quality estimation models are often opaque and computationally expensive, making them impractical to be part of large-scale pipelines. In this work, we tackle two connected challenges: (1) reducing the cost of quality estimation at scale, and (2) developing an inexpensive uncertainty estimation method for quality estimation. To address the latter, we introduce *Instant Confidence COMET*, an uncertainty-aware quality estimation model that matches the performance of previous approaches at a fraction of their costs. We extend this to *Early-Exit COMET*, a quality estimation model that can compute quality scores and associated confidences already at early model layers, allowing us to early-exit computations and reduce evaluation costs. We also apply our model to machine translation reranking. We combine *Early-Exit COMET* with an upper confidence bound bandit algorithm to find the best candidate from a large pool without having to run the full evaluation model on all candidates. In both cases (evaluation and reranking) our methods reduce the required compute by 50% with very little degradation in performance. Finally, we show how *Instant Confidence COMET* can be used to decide which translations a human evaluator should score rather than relying on the COMET score.

1 Introduction

Machine Translation (MT) has made significant progress, with state-of-the-art models achieving increasingly high-quality translations (Kocmi et al., 2023, 2024a). However, as reliance on automatic translation grows, so does the need for robust evaluation metrics. Quality estimation is one such approach, allowing for the automated assessment of translations without the need for reference texts

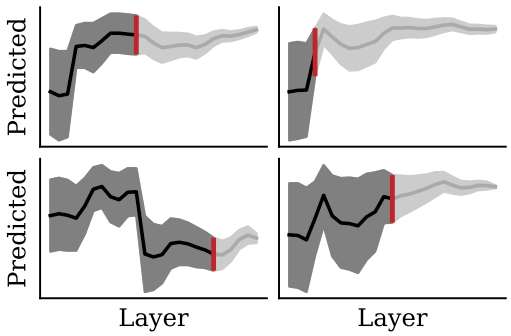

Figure 1: Progression of predicted quality estimation score (dark line) and instant confidence estimation (shaded area) along the quality estimation model computation for four examples from the test set. Layer corresponds to compute cost. Red line | stops computation because the confidence is high enough (early-exit).

(Tamchyna, 2021; Freitag et al., 2023, 2024). Recently, quality estimation models have also been finding their way into the decoding process (Freitag et al., 2022a; Finkelstein et al., 2024), in particular for reranking translation candidates (Shen et al., 2004; Freitag et al., 2022b; Cheng et al., 2024).

Despite its promise, we identify two challenges that prevent the widespread adoption of existing quality estimation methods. First, current methods typically provide a single best-guess quality score, which may be inadequate when uncertainty is high. In critical industry applications, such as legal, medical, or diplomatic contexts, misjudging translation quality can have dire consequences and high scores with very low confidence should better be ignored. Knowing the certainty of a prediction is important for decision-making, such as deferrals or routing (Zhang et al., 2025; Farinhas et al., 2025) or for efficient human evaluation, reserving only uncertain examples for human judgment. Previous work has used techniques such as Monte Carlo dropout (Glushkova et al., 2021) to estimate confidence intervals for quality scores. Although promising, these methods require the model to be run multiple

[†]Code and models: github.com/zouharvi/COMET-early-exit

Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics
Volume 1: Long Papers, pages 55–76

times for each input to generate different stochastic predictions, leading to a substantial computational burden. This limits their practical use in large-scale systems. On the other hand, direct confidence estimation (Zerva et al., 2022) requires separate models for the quality score and the confidence score, which carries a factor two overhead.

Second, the increasing size and complexity of quality estimation models makes it computationally expensive to evaluate hundreds of candidates at once (Guerreiro et al., 2024), severely limiting the scalability. This makes it difficult to use quality estimation in time-sensitive scenarios, where fast evaluations are critical. Additionally, the high computational costs restrict the deployment in resource-constrained environments, hindering practical use in large-scale machine translation.

To overcome the first challenge, we propose *Instant Confidence COMET*, a model that jointly predicts translation quality and an associated uncertainty, without increasing the computational cost. We demonstrate in Section 2 that the model can produce accurate confidence scores that strongly correlate with true prediction errors, without affecting the quality estimation performance.

We also use the confidence mechanism for our new *Early-Exit COMET*, a model that estimates translation quality already in the earlier layers. Examples of how score and confidence predictions develop throughout the layers are shown in Figure 1. Early-Exit COMET uses the confidence score to determine whether an early prediction suffices or additional evaluation layers are needed. In Section 3 we combine it with a simple threshold-based early-exit algorithm for faster sample-level quality estimation. Finally, in Section 4 we use Early-Exit COMET for machine translation reranking. In this task, the machine translation model generates many candidates and the goal is to find the best one with as little compute as possible. We combine Early-Exit COMET with an upper confidence bound bandit algorithm for this reranking process and show that it outperforms strong random and logprob-based baselines.

Another use for confidence scores by *Instant Confidence COMET* is to decide whether a particular translation needs to be scored by humans or if the automated metric suffices. We show positive results for this mixed human-metric annotation in Section 5.

In addition to our results and code, we release the pre-trained Early-Exit and Instant Confidence quality estimation models publicly.

2 Instant Confidence COMET

When a quality estimation model produces an output, such as 85, it is not clear whether this is just the model's best guess or truly an accurate assessment of the quality. Having this additional information is important for decision making (Zhang et al., 2025; Farinhas et al., 2025), such as in commercial translation pipelines where high-scoring translations are marked as requiring very little human attention. If the high score is very uncertain, then it would be safer to have a human double-check these translations. Similarly, when designing budget-efficient human evaluation pipelines (Zouhar et al., 2025b), human labor should be directed to examples where automated metrics are the least confident.

Preliminaries. A quality estimation model is a function f that given a source s and a translation t, computes an estimate of the quality:[1]

$$\widehat{y} = f(s, t) \tag{1}$$

Quality estimation models are usually trained with the $\mathcal{L}_2$ loss between predicted and human assessments of quality (Kocmi et al., 2022, 2024b; Lommel et al., 2014).

$$\mathcal{L}_2(y, f(s, t)) \tag{2}$$

The COMET model (Rei et al., 2020, 2022b) is one such quality estimation function f. The model is based on encoding the source, translation, and potentially also the reference with a multi-layer, multilingual, transformer-based encoder. Once the embeddings are computed, they are joined in a regression head that produces the final score.

For epistemic[2] uncertainty estimation, we wish to have a predictor corresponding to a particular quality estimation model f that predicts the magnitude of the individual sample-level error:

$$\hat{e} = |f(s, t) - y| \tag{3}$$

The negative error then corresponds to confidence or certainty.

[1] If a reference translation r is also included in the input, then the quality score is called a reference-based metric.

[2] Zerva et al. (2022) distinguish between epistemic uncertainty, meaning lack of model knowledge, and aleatoric uncertainty, meaning noise in data.

2.1 Models

We first introduce our instant confidence model and then describe two prior approaches for uncertainty-aware quality estimation.

Instant Confidence COMET. We propose modifying the quality estimation model to output both a quality estimate and an error estimate at the same time. Thus, the model outputs $\langle \hat{y}, \hat{e} \rangle$, and during training we sum two MSE losses with importance of the second determined by hyperparameter β:

$$\mathcal{L}_2(y, \hat{y}) + \beta \cdot \mathcal{L}_2(|y - \hat{y}|, \hat{e}) \quad (4)$$

Notably, we do not backpropagate loss through $\hat{y}$ in the second term, only through $\hat{e}$. The architecture is illustrated in Figure 2. During inference, the model has almost the same computational cost as an unmodified COMET model.

MC Dropout. Glushkova et al. (2021) elicit confidence scores using Monte Carlo (MC) dropout. This involves running the quality estimation model multiple times whilst introducing some randomness, in this case random dropout, and measuring the variance across runs in the output. The underlying hypothesis is that the model outputs, even with dropout, will be the same for high-confidence samples but different for low-confidence samples. While producing strong results, these methods are not practical for real applications because they can require up to 100 model runs.

DUP. Zerva et al. (2022) train a separate secondary model that estimates the error of the original quality estimate. They call this Direct Uncertainty Prediction (DUP). We consider two variants. In the first, the uncertainty predictor does not know the original model prediction and in the second, as described in the previous work, the prediction is passed to the model. Both variants come with a factor two computational overhead. In addition, only the first variant is parallelizable because it does not depend on the original model's prediction.

2.2 Results

We now evaluate the proposed uncertainty-aware quality estimation model and compare it to previous work. We reproduce the previous works under the same conditions to have a level playing field for fair comparisons.

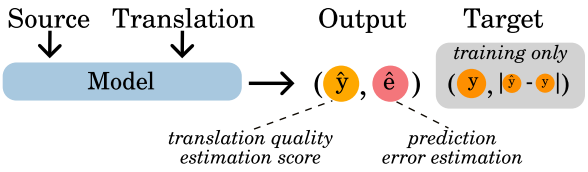

Figure 2: Architecture for uncertainty-aware quality estimation system based on COMET.

	Cost↓	Human↑	Error↑	
Instant Confidence β=0.25	1×	0.316	0.222	
Instant Confidence β=0.5	1×	0.309	0.224	
Instant Confidence β=0.75	1×	0.326	0.228	★
Instant Confidence β=1.0	1×	0.330	0.207	
Instant Confidence β=1.5	1×	0.325	0.200	
No Confidence	1×	0.327	-	
MC Dropout (2)	2×	0.210	0.201	
MC Dropout (5)	5×	0.247	0.267	
MC Dropout (10)	10×	0.262	0.301	
MC Dropout (50)	50×	0.279	0.328	
MC Dropout (100)	101×	0.281	0.333	
MC Dropout† (2)	3×	0.327	0.061	
MC Dropout† (5)	6×	0.327	0.092	
MC Dropout† (10)	11×	0.327	0.115	
MC Dropout† (50)	51×	0.327	0.131	
MC Dropout† (100)	101×	0.327	0.134	
DUP (parallelizable)	2×	0.327	0.135	
DUP (sequential)	2×	0.327	0.216	

Table 1: Correlation (Pearson) of model scores with human scores (Human) and correlation of model error predictions (negative confidence) with true error (Error). Higher is better ↑ and lower is better ↓. MC Dropout† uses dropout only for predicting the error and calculates the quality score without dropout.

Setup. In line with previous works, we use the human-annotated segment-level data from WMT 2019 to 2022 for training (Barrault et al., 2019, 2020; Akhbardeh et al., 2021; Kocmi et al., 2022) and reserve WMT 2023 (Kocmi et al., 2023) for test set (116k segments across 8 language pairs: De↔En, Ja↔En, En↔Zh, Cs→Uk, En→Cs).[3] We describe the general model technical details in Appendix Table 3.

Pointwise confidence is cheap. We compare our approach with previous work in Table 1. The β hyperparameter allows for trade-off between correlation with human score and confidence correlation with true error. For Monte Carlo dropout, as the number of runs is increased, the correlation between the model score variance (model confidence)

[3]We do not use WMT 2024 (Kocmi et al., 2024a) due to the new annotation protocol which leads to lower correlations for all metrics.

and the true error increases. However, this comes at the very steep cost of having to run the same model with dropout multiple times, which is not feasible in many quality estimation applications. Moreover, using dropout during inference hurts the quality estimation performance, because part of the information flow in the network is obscured. Increasing the number of runs only partially compensates for this drop; indeed, even with 100 runs, the score correlation remains lower than that obtained with a single run with no dropout (0.281 vs. 0.327).

Upon first glance, the solution appears to be to run Monte Carlo dropout 100 times to obtain a good uncertainty estimate and once without the dropout to also obtain a good quality estimation. We show this approach as MC Dropout† in Table 1. However, due to the discrepancy between this model score and the uncertainty measure, the error correlation turns out to be very low, revealing this not to be a viable uncertainty prediction setup.

Lastly, we compare with DUP, which has the original (best) human correlation but still twice the cost of our approach and lower error correlation.

Overall, our instant confidence approach is the best in terms of cost and competitive in terms of human and confidence correlations. In Appendix Figure 8 we see that on average when the quality estimation model has high confidence (low $\hat{e}$) then it is likely to have an accurate prediction (low $|\hat{y} - y|$). Notably, the lowest confidence bin corresponds to samples where the quality estimation model is very incorrect ($|\hat{y} - y| > 20$).

3 Early-Exit COMET

COMET is a computationally expensive evaluation method. The 24 transformer layers of the model are costly to compute. The idea of Early-Exit COMET is to use the embeddings from earlier layers to predict the final COMET score in advance. In addition, each layer predicts the error between its score estimate and the full COMET score. In this way, if the model has a confident estimate of the full COMET score after a few layers, then the evaluation can *early-exit* and save on compute costs.

3.1 Early-Exit COMET with Self-Confidence

We make two architectural changes to COMET to enable confidence-aware early-exit: (1) predictions at each layer instead of just after the final layer, and (2) self-confidence predictions that predict the error with respect to the final layer COMET score.[4] Importantly, we do not attempt to predict the error with respect to human scores because we wish to stop if we are confident that the final layer would also not produce a very different output.

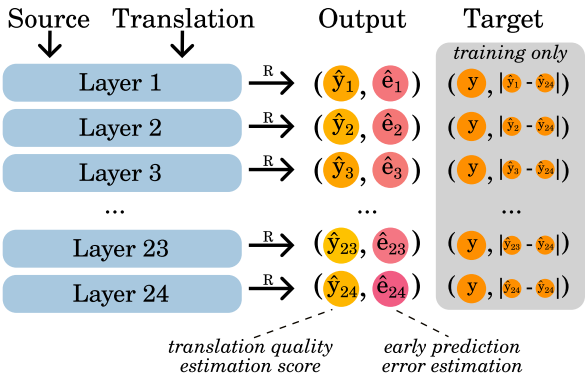

Figure 3: Architecture for confidence-aware (with respect to last layer) early-exit quality estimation system based on COMET.

Let L_i be the embedding after layer i, and let R be the regressor head on top of the final layer. The final model prediction is then:

$$\hat{y} = \mathrm{R}(\mathrm{L}_{|\mathrm{L}|}). \tag{5}$$

We wish to predict approximate scores $\hat{y}_i$ after each layer i. Therefore, instead of training the evaluation model with the standard loss function,

$$\mathcal{L}_2\left(y, \mathrm{R}(\mathrm{L}_{|\mathrm{L}|})\right), \tag{6}$$

we apply the same regressor head to each layer and calculate the cumulative loss:

$$\sum_{i=1}^{|\mathrm{L}|} \mathcal{L}_2\left(y, \mathrm{R}(\mathrm{L}_\mathrm{i})\right). \tag{7}$$

To incorporate confidence predictions, we also increase the output dimensionality of the regressor head to two (R_y and R_e). The second output is used to estimate how far the (early) prediction is from the final prediction. We refer to this as *self-confidence*. This gives an additional loss term for each layer:

$$\mathcal{L}_2\left(\left|\mathrm{R}_y(\mathrm{L}_i) - \mathrm{R}_y(\mathrm{L}_{|\mathrm{L}|})\right|, \mathrm{R}_e(\mathrm{L}_\mathrm{i})\right). \tag{8}$$

The architecture is illustrated in Figure 3.

3.2 Results

We now discuss our findings with Early-Exit COMET. We use the same experimental setup as in Section 2.

[4]This is distinct from Instant Confidence COMET, where the predicted error was with respect to human scores.

Layer \ Layers	01	05	09	13	17	21	24	Human
01	1.00	0.30	0.23	0.17	0.17	0.15	0.15	0.034
05	0.30	1.00	0.93	0.72	0.70	0.66	0.65	0.207
09	0.23	0.93	1.00	0.78	0.75	0.70	0.69	0.221
13	0.17	0.72	0.78	1.00	0.97	0.87	0.85	0.278
17	0.17	0.70	0.75	0.97	1.00	0.91	0.89	0.281
21	0.15	0.66	0.70	0.87	0.91	1.00	0.99	0.312
24	0.15	0.65	0.69	0.85	0.89	0.99	1.00	0.309

Early-Exit COMET

Layer	01	05	09	13	17	21	24	
01	1.00	0.23	0.17	0.10	-0.01	-0.07	-0.06	-0.033
05	0.23	1.00	0.86	0.68	0.48	0.38	0.28	0.064
09	0.17	0.86	1.00	0.90	0.68	0.56	0.41	0.116
13	0.10	0.68	0.90	1.00	0.86	0.73	0.53	0.176
17	-0.01	0.48	0.68	0.86	1.00	0.96	0.73	0.264
21	-0.07	0.38	0.56	0.73	0.96	1.00	0.80	0.283
24	-0.06	0.28	0.41	0.53	0.73	0.80	1.00	0.327

Baseline COMET

Table 2: Pearson correlations between intermediate layer outputs (green) and between intermediate layer outputs and humans (pink) for supervised Early-Exit as described in Section 3.1 (left) and unsupervised Early-Exit based on standard COMET (right). See Appendix Table 5 for detailed version.

Early layer scores. To measure the quality of the early layer scores, we calculate correlations with the final layer scores, as well as with human evaluations. For comparison, we include the baseline COMET model, applying the final layer regressor to the intermediate embeddings to get intermediate scores. In Table 2 we show that earlier layer scores of baseline COMET do not correlate strongly with final layer scores or with human judgments. However, with direct supervision at each layer, we see much better results for Early-Exit COMET. At layer 5 we already see a correlation score of 0.65 with the final layer and 0.207 with human scores. By layer 13, the correlation with human scores is 0.278, comparable to the final layer. We include a version of Early-Exit COMET with separate regression heads for each layer in Appendix Table 5, but we do not observe any improvements.

To measure the quality of the self-confidence error predictions, we plot the average predicted error versus the true error in Appendix Figure 7. We also include correlation scores for selected layers showing the correlation between the predicted and true errors, e.g., 0.44 for layer 9 scores. This enables early-exit decision making, which we introduce in the next section.

3.3 Deciding When to Early-Exit

In some cases, the Early-Exit COMET model is already close to the final assessment after a few layers (top row in Figure 1). In these cases, we do not need to continue the computation if we are confident that the final outcome will be very close. The per-layer confidences can inform this decision.

Inputs: Source s, translation t, threshold τ
Output: Quality estimate $\hat{y}$

Compute $L_0(s,t)$
for $i \in 1 \dots |L|$ **do**
  Compute $L_i(s,t)$ from $L_{i-1}(s,t)$ ▷ next layer
  $\hat{y}_i, \hat{e}_i \leftarrow R(L_i)$ ▷ apply regressor head
  if $\hat{e}_i < \tau$ **then return** $\hat{y}_i$ ▷ early-exit
end for
return $\hat{y}_{|L|}$

Algorithm 1: Confidence-Exit with Early-Exit COMET.

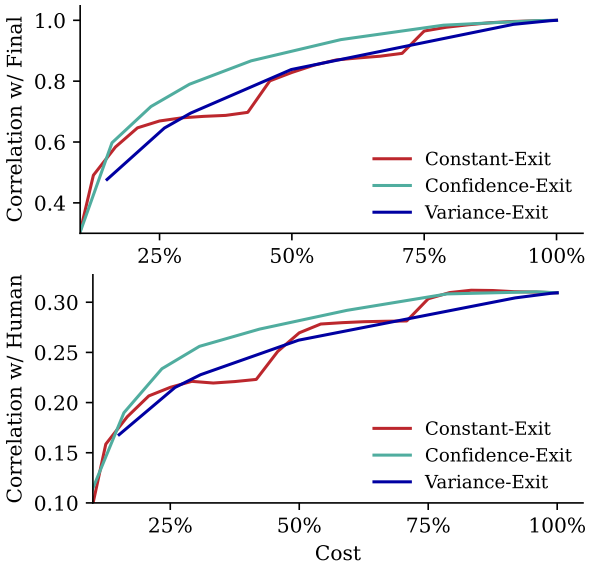

Figure 4: Quality estimation correlation with last layer (final prediction, top) and with human scores (bottom) for heuristic-based early-exit COMET.

In Algorithm 1, we outline a simple heuristic that stops the Early-Exit COMET computation when the error prediction is low enough. To evaluate this algorithm, we compare it with two baselines that do not use the confidence scores: (1) Constant-Exit (Appendix, Algorithm 3) stops at a constant, predefined layer; and (2) Variance-Exit (Appendix, Algorithm 4) stops when the variance of three consecutive predictions is under a chosen threshold.

Results. Figure 4 shows the performance of Algorithm 1 versus the baselines. We plot the correlation with final layer scores and human judgments for different budgets (total computation costs). The cost is relative to calculating the full COMET score for all inputs (100%). We vary the algorithm thresholds to explore different budgets.

We see that the confidence-based early-exit algo-

Inputs: Early-Exit model with predictions $\hat{y}_l$ and confidences (error estimates) e_l for layer $l \in [1, |L|]$, translation candidates $\mathcal{C}$, exploration-exploitation hyperparameter γ, total evaluation budget B
Output: final translation candidate $c^* \in \mathcal{C}$

$\sigma \leftarrow e \cdot \sqrt{\frac{\pi}{2}}$ ▷ Rescale the absolute error estimates (MAE) to standard deviation estimates
$S_\mathcal{C} \leftarrow \{(\hat{y}_1(c), \sigma_1(c)) | c \in \mathcal{C}\}$ ▷ Calculate and cache first layer COMET for all candidates
$\hat{B} \leftarrow |\mathcal{C}|$ ▷ Initialize running evaluation costs
$\mathcal{C}' \leftarrow \mathcal{C}; \quad l(c) \leftarrow 1 \; \forall c \in \mathcal{C}'$ ▷ Initialize remaining candidates and explored layers
while $\hat{B} < B$ **and** $\mathcal{C}' \neq \emptyset$ **do**
  $\text{UCB}(c) \leftarrow \hat{y}_{l(c)}(c) + \gamma\sigma_{l(c)}(c) \quad \forall c \in \mathcal{C}'$ ▷ Calculate UCB for all (remaining) candidates
  $c^* \leftarrow \arg\max_{c \in \mathcal{C}'} \text{UCB}(c)$ ▷ Choose candidate with highest UCB
  $l(c^*) \leftarrow l(c^*) + 1; \quad \hat{B} \leftarrow \hat{B} + 1$
  $S_\mathcal{C} \leftarrow S_\mathcal{C} \cup (\hat{y}_{l(c^*)}(c^*), \sigma_{l(c^*)}(c^*))$ ▷ Calculate next layer COMET score for c^*
  if $l(c^*) = |L|$ **then**
    $\mathcal{C}' \leftarrow \mathcal{C}' \setminus \{c^*\}$ ▷ Remove c^* if fully evaluated
  end if
end while
return $c^* \leftarrow \arg\max_{c \in \mathcal{C}}(\hat{y}_{l(c)}(c))$ ▷ Return best candidate based on the most advanced (highest layer) predictions for each candidate.

Algorithm 2: Upper Confidence Bound (UCB) Bandit for selecting a high scoring translation candidate from a pool of candidates using Early-Exit COMET.

rithm outperforms variance-exit and constant-exit. This shows that the confidence outputs at early layers are crucial for enabling early-exit. With only half of the compute, we see only a small performance drop from the full to half compute: (0.309→0.292) for human scores.

4 Early-Exit COMET for Reranking

In addition to enhancing the speed of quality estimation, the early-exit model can be applied to reranking. In this setup, a machine translation model generates a set of candidate translations, $\mathcal{C}$, for a source sentence, and the objective of reranking is to identify the best candidate from $\mathcal{C}$.

Reranking with quality estimation has been shown to improve translation quality (Freitag et al., 2022b) and, as one might expect, the larger the initial pool of candidates, the higher the final translation quality (Vernikos and Popescu-Belis, 2024). However, running a quality estimation model on a large number of candidates, e.g. $|\mathcal{C}| > 100$, can be prohibitively expensive.

To lower these compute costs, we turn to Early-Exit COMET and rely on its accurate early-layer predictions to make reranking more efficient. The idea is to calculate more accurate (and more expensive) scores only for the most promising candidates based on the less expensive early-layer scores.

Each early-exit output, $\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|L|}$, has an associated runtime cost: $\text{cost}(\hat{y}_i)$. For Early-Exit COMET, we can take $\text{cost}(\hat{y}_i) = i$ since each layer has the same computation cost. Note that costs are not additive; for example, once the layer 3 score is calculated, all later layers can use the layer 3 embeddings. So for $i > 3$, we will now only accumulate the additional cost of $\text{cost}(\hat{y}_i) - \text{cost}(\hat{y}_3)$.

Ultimately, we must select a final candidate and our goal is to strike a balance between the total computation cost and finding the candidate with the highest $\hat{y}_{|L|}$ score.

4.1 Upper Confidence Bound Bandit

In the multi-armed bandit problem, a decision-maker must repeatedly choose between several actions, "arms", with unknown reward distributions. The goal is to maximize the cumulative reward earned over time by balancing exploration (trying out different arms to learn their rewards) and exploitation (choosing the arm believed to be the best), which can be done with the upper confidence bound bandit (Auer, 2002). The algorithm computes an upper confidence bound for the estimated reward for each arm, selecting the arm with the highest bound in each round. This approach encourages the algorithm to explore less-pulled arms (with larger error estimates) while exploiting those with higher estimated rewards.

In our context, the "arms" are the translation candidates $\mathcal{C}$, and pulling an arm corresponds to calculating an additional quality estimation model layer. Since the predictions improve with more layers (Table 2), each "pull" of the "arm" improves our estimate of the reward for the given candidate. For each candidate, we always consider the reward of the highest layer explored thus far and its associated confidence score. The computation budget determines the number of pulls available. When the budget runs out, we pick the candidate with the highest reward estimate, ignoring the associated

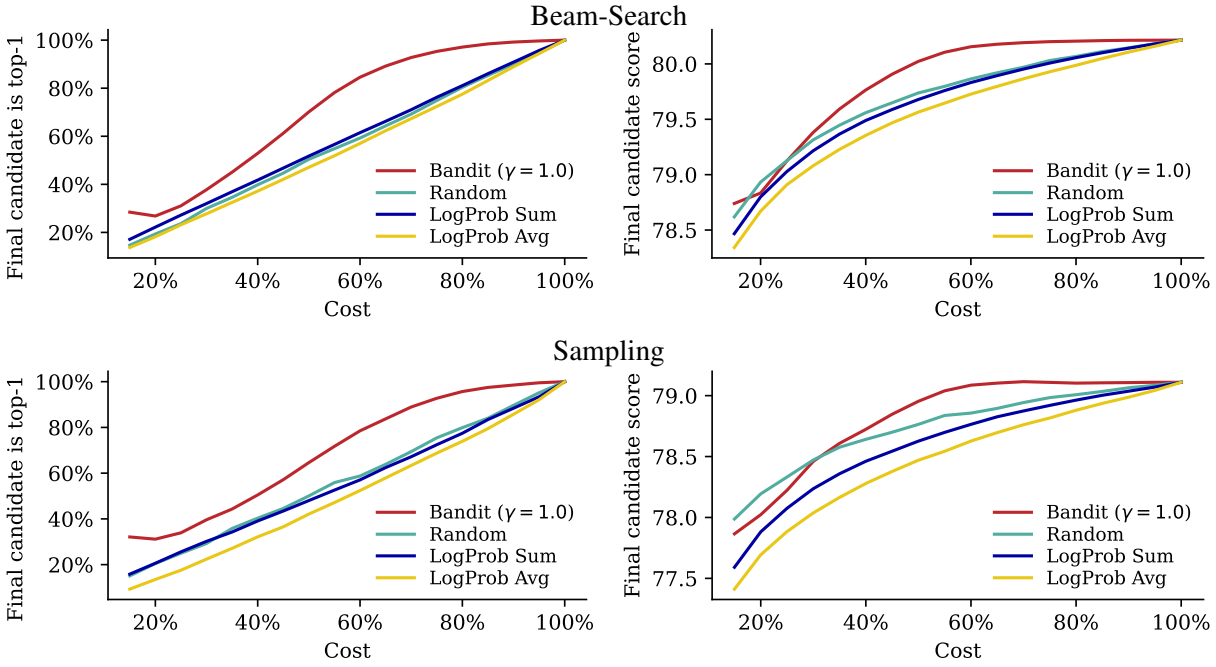

Figure 5: Quality of the candidates returned by the Upper Confidence Bound bandit. Quality is measured in terms of the average final candidate score and the proportion to top-1 candidates selected. We plot these measures for various evaluation budgets. Cost (or budget) is given relative to calculating the full COMET scores for all candidates (100%). See results in tabular form in Appendix Table 4.

confidence scores. A more precise formulation is provided in Algorithm 2. The γ controls the balance between exploration and exploitation. We use $\gamma = 1$ for most experiments, but an ablation study can be found in Appendix E.

Setup. To evaluate our upper confidence bound bandit with Early-Exit COMET, we again use the WMT 2023 test set (Kocmi et al., 2023), including 8 language pairs[5]. We generate translation candidates using NLLB (Team et al., 2022), with 200 candidates per segment. We generate candidates via multinomial sampling (across the whole vocabulary) and via beam search separately and report results for both.

Baselines. We compare against a *Random* baseline, where we select a random subset of candidates, calculate the full COMET scores for this subset and select the candidate with the highest score. The size of the subset is proportional to the budget. In addition to the random baseline, we also show results for two baselines based on the log probability scores (logprobs) of the translation model (NLLB). In contrast to the random baseline, we select the subset based on the logprobs of the candidate tokens. We take the highest-scoring candidates in terms of the average logprob score (*LogProb Avg*) or the sum of logprob scores (*LogProb Sum*), and calculate the full COMET scores for this subset.

[5]Due to the large size, we use a random subset of the test data with 2000 source examples for each language pair.

4.2 Results

Figure 5 shows the quality of the bandit output as we increase the evaluation budget. We plot both the average score of the final candidates and the rate at which the top-1 (best full COMET score) candidate is selected.

Surprisingly, the logprob-based baselines underperform the random baseline in almost all settings, in both metrics, and for all budgets. We hypothesize that this is due to lower diversity in the selected candidates subset. In contrast, the bandit outperforms the random baseline in almost all scenarios and for all budgets. In particular, with 60% of the compute budget, there is almost no drop in the translation quality. This indicates that (1) the Early-Exit COMET scores and error predictions are valuable and (2) the upper confidence bound bandit makes efficient use of these estimates.

For more detailed results on the upper confidence bound bandit, please see the ablation studies in Appendix E.

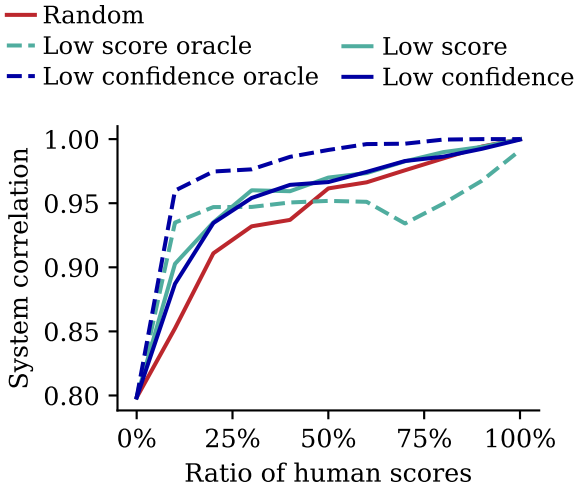

Figure 6: System-level correlation (macro Spearman) of a combined human-metric annotation with part of segments being annotated by humans. At 0%, all segments are scored by the metric. At 100%, all segments are scored by humans.

5 Deferring to Humans

In this section, we use the confidence scores from Instant Confidence COMET to decide whether we can rely directly on the COMET score, or whether a translation should instead be scored by a human annotator.

Methods. As the simplest baseline, we consider randomly selecting examples for human annotation. For a stronger baseline, as used in Zouhar et al. (2025a), we select the examples with the lowest scores for human annotation. This can be done with existing metrics. We compare this with our approach of human-annotating examples with the lowest confidence scores (highest predicted error). Finally, for both the low-score and low-confidence approaches, we also consider oracle versions where we prioritize examples with low human scores and high absolute error respectively.

Setup. We analyze deferral rates ranging between 0% and 100%. To measure the success of the combined human-metric annotation, we look at how the final ranking of the machine translation model changes degrades as we score more of the data with the automated metric. We quantify this by the system-level Spearman correlation between ranking based on the combined annotation approach and 100% human annotation. We macro-average results across all considered languages in the dataset.

Results. The results are shown in Figure 6. Our method outperforms random selection, though is on par with deferring low-scoring examples. However, lower scores correlate (0.243 macro Pearson) with the number of words in the translation. The length of the translation affects the human annotation time (Zouhar et al., 2025a) and thus e.g. 20% deferral with this approach might use more annotation budget than 20% deferral with random selection. On the other hand, low confidence does not correlate meaningfully (-0.087 macro Pearson) with the translation length, thus making it a safer option.

Finally, we note that there is a large gap between using the predicted confidence scores and the oracle confidence. This means, that improvements in estimating the automated metric error can lead to further gains in deferral efficiency.

6 Related Work

We now position our contributions in relation to other works on uncertainty-aware quality estimation, faster inference, and reranking.

Model uncertainty. Quantifying uncertainty in learned automated machine translation metrics was first proposed by Glushkova et al. (2021), who use the variance from ensembles or Monte Carlo dropout as a measure of uncertainty. A different approach by Zerva et al. (2022) introduces a secondary confidence estimation model to complement the original quality estimation model.[6] Instead, we optimize a single model that produces two scores: a quality estimate and a measure of uncertainty. We optimize this model jointly in each training step, optionally optimizing for predictions at each layer and optionally with respect to the last layer's prediction. Finally, we note that the focus of Zerva et al. (2022) is on analyzing the source of uncertainty, while our focus is on using uncertainty to make quality estimation more reliable and efficient. We direct the reader to the work of Lahlou et al. (2023) for more theoretical yet comprehensive treatment of directly estimating model confidence.

Faster quality estimation. Multiple previous works investigated improving the efficiency of calculating trained metrics. For large scales, Rei et al. (2022a) use length-batching to speed up inference. At the same time, they statically prune the quality

[6] They also propose an approach whereby one model predicts both quantities, similar to our instant confidence. See Appendix B documenting our attempts at replicating this model.

estimation model, which is similar to our constant-exit approach. Cheng et al. (2024) use a smaller baseline language model rather than the default XLM-Roberta for COMET. Zouhar et al. (2024) find that simply quantizing the model to half precision has almost no effect on the final quality estimation performance while halving the compute costs. Gowda et al. (2023, 2024) port COMET to a faster inference engine for massive speed gains. Larionov et al. (2024) explore pruning, distillation, and quantization for a very large quality estimation model, xCOMET (Guerreiro et al., 2024). All of these approaches are orthogonal to our method and could be used in combination.

Early-Exit. Many works have explored intermediate model predictions (Liu et al., 2019; Belrose et al., 2023, inter alia). However, the key ingredient is to know when to stop the computation, such as at a particular Transformer block layer (Zhou et al., 2020), but these methods are largely applicable to classification tasks. Xin et al. (2021) propose *learning-to-exit*, which we loosely follow in our work. However, instead of predicting a probability of success, we predict the absolute error of the model, which is directly interpretable.

Reranking. Reranking improves translation quality (Freitag et al., 2022b), but scoring large candidate sets is computationally expensive. One common approach is minimum Bayes risk (MBR) decoding (MBR; Eikema and Aziz, 2020), which selects the translation candidate with the lowest expected risk. Recent work has made MBR more efficient (Cheng and Vlachos, 2023; Deguchi et al., 2024; Trabelsi et al., 2024; Vamvas and Sennrich, 2024), including methods that pre-select candidates with cheaper, noisier scoring functions (Fernandes et al., 2022; Eikema and Aziz, 2022).

Other approaches improve efficiency through token-level reranking (Singhal et al., 2023) or by framing reranking as a Bayesian Optimization problem, where a cheaper scoring model assists in identifying high-quality candidates before applying the more expensive scoring model (Cheng et al., 2024).

7 Conclusion

We introduced three approaches to improve the efficiency of quality estimation. Our instant confidence Early-Exit COMET achieves comparable performance to Monte Carlo dropout methods while drastically lowering computational overhead. Combining our model with a simple early-exit strategy, we can compute comparable quality estimation scores without having to compute the full quality estimation model. Combining Early-Exit COMET with an upper confidence bound bandit, we speed up candidate reranking for machine translation by a factor of almost 2 with negligible impact on translation quality. Finally, the confidence scores can also inform which translations to human-annotate.

Recommendations. Based on our findings, we offer the following practical advice:

- When quality estimation is part of a more complex decision process, we recommend using instant confidence-aware COMET to provide additional information on the credibility of decisions.
- For very large-scale quality estimation use cases with limited compute budget, we recommend Early-Exit COMET with Confidence-Exit.
- For reranking with very large candidate pools, we recommend the upper confidence bound bandit to reduce the number of scored candidates.
- When human-annotating only a portion of translations, prioritize those with low Instant Confidence COMET confidence.

Future work. Reranking can also be combined with beam search to improve quality at generation time. Future work could apply our ideas to improve reranking efficiency for model generation. Moreover, in Appendix G we describe a prototype of Partial COMET, a quality estimation model that is able to robustly evaluate incomplete generations. This can be used to prune incomplete candidates, thus saving unnecessary computation of the very expensive generative model.

Limitations

Regarding the results in Table 1, it is possible that there is a causal trade-off between the two correlation scores, the correlation with human scores and the correlation with the true error. For example, it could be easier to predict the confidence of a model that performs worse. However, making stronger claims would require a more thorough mathematical treatment, which is outside of the focus of this work.

Ethics Statement

Data used in this paper was collected by previous works. The authors foresee no ethical problems.

Acknowledgments

We thank the EAMT committee for sponsoring this research. We thank the Vector Stiftung for supporting Béni Egressy's work. Part of this work received support from the European Union's Horizon research and innovation programme under grant agreement No 101135798, project Meetween (My Personal AI Mediator for Virtual MEETtings BetWEEN People). This research has been funded in part by a Swiss National Science Foundation award (project 201009) and a Responsible AI grant by the Haslerstiftung.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina EspañBonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, 1–88, Online. Association for Computational Linguistics.

Chantal Amrhein and Barry Haddow. 2022. Don't discard fixed-window audio segmentation in speech-to-text translation. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 203–219. Association for Computational Linguistics.

Peter Auer. 2002. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 3:397–422.

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 conference on machine translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, 1–55. Association for Computational Linguistics.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, 1–61. Association for Computational Linguistics.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens.

Julius Cheng and Andreas Vlachos. 2023. Faster minimum Bayes risk decoding with confidence-based pruning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 12473–12480. Association for Computational Linguistics.

Julius Cheng, Maike Züfle, Vilém Zouhar, and Andreas Vlachos. 2024. A bayesian optimization approach to machine translation reranking.

Hiroyuki Deguchi, Yusuke Sakai, Hidetaka Kamigaito, Taro Watanabe, Hideki Tanaka, and Masao Utiyama. 2024. Centroid-based efficient minimum Bayes risk decoding. In *Findings of the Association for Computational Linguistics: ACL 2024*, 11009–11018. Association for Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2020. Is MAP decoding all you need? the inadequacy of the mode in neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 4506–4520. International Committee on Computational Linguistics.

Bryan Eikema and Wilker Aziz. 2022. Sampling-based approximations to minimum Bayes risk decoding for neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 10978–10993. Association for Computational Linguistics.

António Farinhas, Nuno M. Guerreiro, Sweta Agrawal, Ricardo Rei, and André F. T. Martins. 2025. Translate smart, not hard: Cascaded translation systems with quality-aware deferral.

Patrick Fernandes, António Farinhas, Ricardo Rei, José G. C. de Souza, Perez Ogayo, Graham Neubig, and Andre Martins. 2022. Quality-aware decoding for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 1396–1412. Association for Computational Linguistics.

Mara Finkelstein, Subhajit Naskar, Mehdi Mirzazadeh, Apurva Shah, and Markus Freitag. 2024. MBR and QE finetuning: Training-time distillation of the best and most expensive decoding methods.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022a. High quality rather than high model probability: Minimum bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Markus Freitag, David Grangier, Qijun Tan, and Bowen Liang. 2022b. High quality rather than high model probability: Minimum Bayes risk decoding with neural metrics. *Transactions of the Association for Computational Linguistics*, 10:811–825.

Markus Freitag, Nitika Mathur, Daniel Deutsch, Chi-Kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Frederic Blain, Tom Kocmi, Jiayi Wang, David Ifeoluwa Adelani, Marianna Buchicchio, Chrysoula Zerva, and Alon Lavie. 2024. Are LLMs breaking MT metrics? results of the WMT24 metrics shared task. In *Proceedings of the Ninth Conference on Machine Translation*, 47–81. Association for Computational Linguistics.

Markus Freitag, Nitika Mathur, Chi-kiu Lo, Eleftherios Avramidis, Ricardo Rei, Brian Thompson, Tom Kocmi, Frederic Blain, Daniel Deutsch, Craig Stewart, Chrysoula Zerva, Sheila Castilho, Alon Lavie, and George Foster. 2023. Results of WMT23 metrics shared task: Metrics might be guilty but references are not innocent. In *Proceedings of the Eighth Conference on Machine Translation*, 578–628. 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: EMNLP 2021*, 3920–3938. Association for Computational Linguistics.

Thamme Gowda, Roman Grundkiewicz, Elijah Rippeth, Matt Post, and Marcin Junczys-Dowmunt. 2024. PyMarian: Fast neural machine translation and evaluation in python. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 328–335. Association for Computational Linguistics.

Thamme Gowda, Tom Kocmi, and Marcin Junczys-Dowmunt. 2023. Cometoid: Distilling strong reference-based machine translation metrics into Even stronger quality estimation metrics. In *Proceedings of the Eighth Conference on Machine Translation*, 751–755. Association for Computational Linguistics.

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.

HyoJung Han, Kevin Duh, and Marine Carpuat. 2024. SpeechQE: Estimating the quality of direct speech translation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 21852–21867. Association for Computational Linguistics.

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*, 492–504. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Marzena Karpinska, Philipp Koehn, Benjamin Marie, Christof Monz, Kenton Murray, Masaaki Nagata, Martin Popel, Maja Popović, Mariya Shmatova, Steinthór Steingrímsson, and Vilém Zouhar. 2024a. Findings of the WMT24 general machine translation shared task: The LLM era is here but MT is not solved yet. In *Proceedings of the Ninth Conference on Machine Translation*, 1–46. Association for Computational Linguistics.

Tom Kocmi, Eleftherios Avramidis, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Markus Freitag, Thamme Gowda, Roman Grundkiewicz, Barry Haddow, Philipp Koehn, Benjamin Marie, Christof Monz, Makoto Morishita, Kenton Murray, Makoto Nagata, Toshiaki Nakazawa, Martin Popel, Maja Popović, and Mariya Shmatova. 2023. Findings of the 2023 conference on machine translation (WMT23): LLMs are here but not quite there yet. In *Proceedings of the Eighth Conference on Machine Translation*, 1–42. Association for Computational Linguistics.

Tom Kocmi, Rachel Bawden, Ondřej Bojar, Anton Dvorkovich, Christian Federmann, Mark Fishel, Thamme Gowda, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Rebecca Knowles, Philipp Koehn, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Michal Novák, Martin Popel, and Maja Popović. 2022. Findings of the 2022 conference on machine translation (WMT22). In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 1–45. Association for Computational Linguistics.

Tom Kocmi, Vilém Zouhar, Eleftherios Avramidis, Roman Grundkiewicz, Marzena Karpinska, Maja Popović, Mrinmaya Sachan, and Mariya Shmatova. 2024b. Error span annotation: A balanced approach for human evaluation of machine translation. In *Proceedings of the Ninth Conference on Machine Translation*, 1440–1453. Association for Computational Linguistics.

Salem Lahlou, Moksh Jain, Hadi Nekoei, Victor Ion Butoi, Paul Bertin, Jarrid Rector-Brooks, Maksym Korablyov, and Yoshua Bengio. 2023. DEUP: Direct epistemic uncertainty prediction.

Daniil Larionov, Mikhail Seleznyov, Vasiliy Viskov, Alexander Panchenko, and Steffen Eger. 2024. xCOMET-lite: Bridging the gap between efficiency and quality in learned mt evaluation metrics.

Nelson F. Liu, Matt Gardner, Yonatan Belinkov, Matthew E. Peters, and Noah A. Smith. 2019. Linguistic knowledge and transferability of contextual

representations. 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)*, 1073–1094. Association for Computational Linguistics.

Arle Lommel, Hans Uszkoreit, and Aljoscha Burchardt. 2014. Multidimensional quality metrics (mqm): A framework for declaring and describing translation quality metrics. *Tradumàtica*, 0455–463.

Ahmed Magooda and Cezary Marcjan. 2020. Attend to the beginning: A study on using bidirectional attention for extractive summarization.

Ricardo Rei, Ana C Farinha, José G.C. de Souza, Pedro G. Ramos, André F.T. Martins, Luisa Coheur, and Alon Lavie. 2022a. Searching for COMETINHO: The little metric that could. In *Proceedings of the 23rd Annual Conference of the European Association for Machine Translation*, 61–70. European Association for Machine Translation.

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)*, 2685–2702. 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. 2022b. CometKiwi: IST-unbabel 2022 submission for the quality estimation shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, 634–645. Association for Computational Linguistics.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. Evaluating multilingual speech translation under realistic conditions with resegmentation and terminology. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*, 62–78. Association for Computational Linguistics.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, 177–184. Association for Computational Linguistics.

Prasann Singhal, Jiacheng Xu, Xi Ye, and Greg Durrett. 2023. EEL: Efficiently encoding lattices for reranking. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 9299–9316. Association for Computational Linguistics.

Matthias Sperber, Ondřej Bojar, Barry Haddow, Dávid Javorský, Xutai Ma, Matteo Negri, Jan Niehues, Peter Polák, Elizabeth Salesky, Katsuhito Sudoh, and Marco Turchi. 2024. Evaluating the IWSLT2023 speech translation tasks: Human annotations, automatic metrics, and segmentation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, 6484–6495. ELRA and ICCL.

Aleš Tamchyna. 2021. Deploying MT quality estimation on a large scale: Lessons learned and open questions. In *Proceedings of Machine Translation Summit XVIII: Users and Providers Track*, 291–305. Association for Machine Translation in the Americas.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation.

Firas Trabelsi, David Vilar, Mara Finkelstein, and Markus Freitag. 2024. Efficient minimum bayes risk decoding using low-rank matrix completion algorithms. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jannis Vamvas and Rico Sennrich. 2024. Linear-time minimum Bayes risk decoding with reference aggregation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 790–801. Association for Computational Linguistics.

Giorgos Vernikos and Andrei Popescu-Belis. 2024. Don't rank, combine! combining machine translation hypotheses using quality estimation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 12087–12105. Association for Computational Linguistics.

Ji Xin, Raphael Tang, Yaoliang Yu, and Jimmy Lin. 2021. BERxiT: Early exiting for BERT with better fine-tuning and extension to regression. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 91–104. Association for Computational Linguistics.

Chrysoula Zerva, Taisiya Glushkova, Ricardo Rei, and André F. T. Martins. 2022. Disentangling uncertainty in machine translation evaluation. In *Proceedings of*

the 2022 Conference on Empirical Methods in Natural Language Processing, 8622–8641. Association for Computational Linguistics.

Tuo Zhang, Asal Mehradfar, Dimitrios Dimitriadis, and Salman Avestimehr. 2025. Leveraging uncertainty estimation for efficient llm routing.

Wangchunshu Zhou, Canwen Xu, Tao Ge, Julian McAuley, Ke Xu, and Furu Wei. 2020. Bert loses patience: fast and robust inference with early exit. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Vilém Zouhar, Pinzhen Chen, Tsz Kin Lam, Nikita Moghe, and Barry Haddow. 2024. Pitfalls and outlooks in using COMET. In *Proceedings of the Ninth Conference on Machine Translation*, 1272–1288. Association for Computational Linguistics.

Vilém Zouhar, Tom Kocmi, and Mrinmaya Sachan. 2025a. AI-assisted human evaluation of machine translation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 4936–4950, Albuquerque, New Mexico. Association for Computational Linguistics.

Vilém Zouhar, Peng Cui, and Mrinmaya Sachan. 2025b. How to select datapoints for efficient human evaluation of nlg models?

A Model Architecture and Training Details

Table 3 provides details of the model architecture and training hyperparameters that were used to train the COMET models in this paper.

Unless otherwise specified, the configuration of referenceless COMET is used. Our models are largely compatible with the upstream COMET repository and can be reproduced based on our code. We run all our experiments on Nvidia A100 (40GB) GPUs, taking about 8 hours to train a single model (2.2GB) for 5 epochs. For each setting, we train a single model and report its performance. In total, all experiments, including preliminary ones, amounted to approximately 20×8 hours = 160 hours of compute on the aforementioned GPU. We base our experiments on a modified COMET v2.2.4 codebase with other package versions listed as dependencies in this version.

Encoder	xlm-roberta-large (24 layers)
Embeddings	Layerwise attention & CLS
Encoder frozen	30% of first epoch
Regression head	$(4\times768) \times 2048\times1024 \times (1 \text{ or } 2)$
Optimizer	AdamW
Learning rate	1.5×10^{-5}, encoder 10^{-6}
Batch size	256 (simulated)
Loss	MSE for both targets
Training epochs	5

Table 3: Model architecture and training details.

B Replicating HTS model

Zerva et al. (2022) propose a quality estimation model that outputs a distribution by predicting its mean and variance. Loosely, this can be interpreted as the score and confidence prediction. However, no public model is available and we have been unable to reproduce the model based on the publicly available code. Upon making the changes in the code that make the codebase compatible with up-to-date packages we did train a model with the `hts` loss, though the resulting human and error correlations were only 0.247 and 0.206, respectively. Because this strays far from the original reported results, we attribute this to reproducibility failure as opposed to failure of the method. Zouhar et al. (2024) already show that the differences in COMET codebase versions can cause large discrepancies in COMET model behavior. While our work uses the latest COMET v2.2.4, the work of Zerva et al. (2022) used COMET v1.0.0rc4.

C Additional Plots for Instant Confidence COMET

We provide additional plots showing the quality of the Instant Confidence COMET error predictions. In Figure 7 we plot the average predicted error versus the true error to indicate average alignment. We also include correlation scores for selected layers showing the correlation between the predicted and true errors, e.g., 0.44 for layer 9 scores. In Figure 8 we plot the average error for different confidence bins based on Instant Confidence. We see that the true error decreases as the predicted confidence increases. The plot also indicates that the score predictions with the highest and lowest true errors are reliably identified by the predicted instant confidence values.

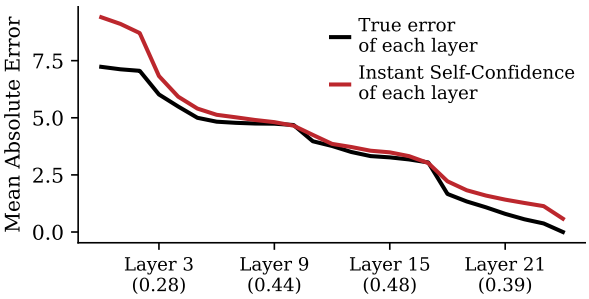

Figure 7: Correspondence of true and instant self-confidence. Correlations in brackets are Pearson correlation for each layer.

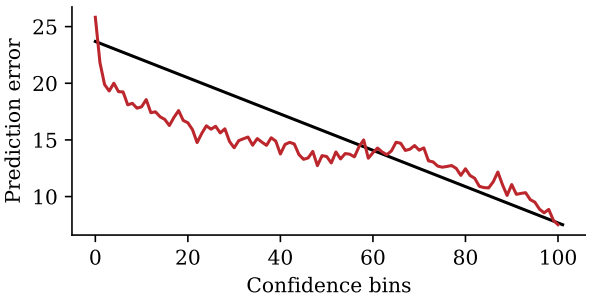

Figure 8: Calibration of predicted quality estimation model confidence based on 100 confidence bins (x-axis) and mean true absolute error of the prediction in each bin (y-axis).

D Baseline Early Exit Algorithms

Below we provide pseudocode for the early-exit algorithms that we use as baselines to compare with Confidence-Exit. These baselines do not use the confidence scores of Early-Exit COMET.

Algorithm 3 exits the COMET evaluation after a fixed, predefined layer, exit layer k.

Algorithm 4 exits the COMET evaluation when Early-Exit COMET scores from three consecutive layers are close to each other, or more precisely, have variance below a chosen threshold τ.

Inputs: Source s, translation t, exit layer k
Output: Quality estimation $\hat{y}$

Compute $L_0(s,t)$
for $i \in 1 \dots k$ **do**
  Compute $L_i(s,t)$ from $L_{i-1}(s,t)$ ▷ next layer
end for
$\hat{y}_k, \hat{e}_k \leftarrow R(L_k)$ ▷ apply regressor head
return $\hat{y}_k$

Algorithm 3: Constant-Exit with Early-Exit COMET.

Inputs: Source s, translation t, threshold τ
Output: Quality estimation $\hat{y}$

Compute $L_0(s,t)$
for $i \in 1 \dots |L|$ **do**
  Compute $L_i(s,t)$ from $L_{i-1}(s,t)$ ▷ next layer
  $\hat{y}_i, \hat{e}_i \leftarrow R(L_i)$ ▷ apply regressor head
  if $\mathrm{Var}[\hat{y}_{i-2:i}] < \tau$ **then return** $\hat{y}_i$ ▷ early-exit
end for
return $\hat{y}_{|L|}$

Algorithm 4: Variance-Exit with Early-Exit COMET.

E Upper Confidence Bound Bandit Ablations

We now describe two variations to the Upper Confidence Bound Bandit algorithm described in Section 4: (1) starting at later layers, and (2) balancing exploration and exploitation.

Starting at Different Layers. Given the significant jump in COMET score accuracy within the first few layers of the COMET model (Table 2), we decided to explore initializing the algorithm with different starting layers. This carries a higher initial cost as we have to run the first few layers of the COMET model for all candidates but could lead to better-informed exploration with the remaining budget. The results in Figure 9 show that this leads to only marginal improvements.

Exploration-Exploitation Tradeoff. The heart of the multi-armed bandit problem is the exploration-exploitation trade-off. In our algorithm this trade-off is controlled by the hyperparameter γ. The higher one chooses γ, the higher the Upper Confidence Bound scores for uncertain candidates will be, and therefore the more likely the algorithm will be to explore many candidates. On the other hand, a low γ will lead the algorithm to go deep with the most promising candidates, i.e., those with the highest estimated scores. We provide results for two different values for γ in Figure 10, which shows that the default choice is likely the most apt.

Distribution of Max. Layers Calculated. We analyze the order in which the layers were calculated by the UCB bandit algorithm by taking snapshots of the max. layer calculated across all candidates at 5% budget increments, from 5% to 100%. We provide plots for the default UCB bandit hyperparameter values (start layer = 1, $\gamma = 1.0$) on the test set with the candidates generated via sampling.

At the start (budget, $B = 5\%$), only the first layer COMET scores are explored for (almost) all candidates, and by the end (budget, $B = 100\%$) the full COMET scores are calculated for all candidates. However, the distributions in between reveal that the bandit often explores the candidates up until certain modal layers. These can be clearly seen (e.g. for budget $B = 50\%$) to be layers 3, 4, 11, 18, and 24. Looking at Figure 7, it can be seen that these values correspond to layers where the predicted error rates (uncertainties) drop significantly, meaning that the evaluation model is now more certain about these scores relative to scores for other candidates. This explains why the UCB bandit algorithm would then prefer to explore less certain candidates with higher UCB scores.

F Reranking Results in Tabular Format

Table 4 provides the results from Figure 5 in tabular format.

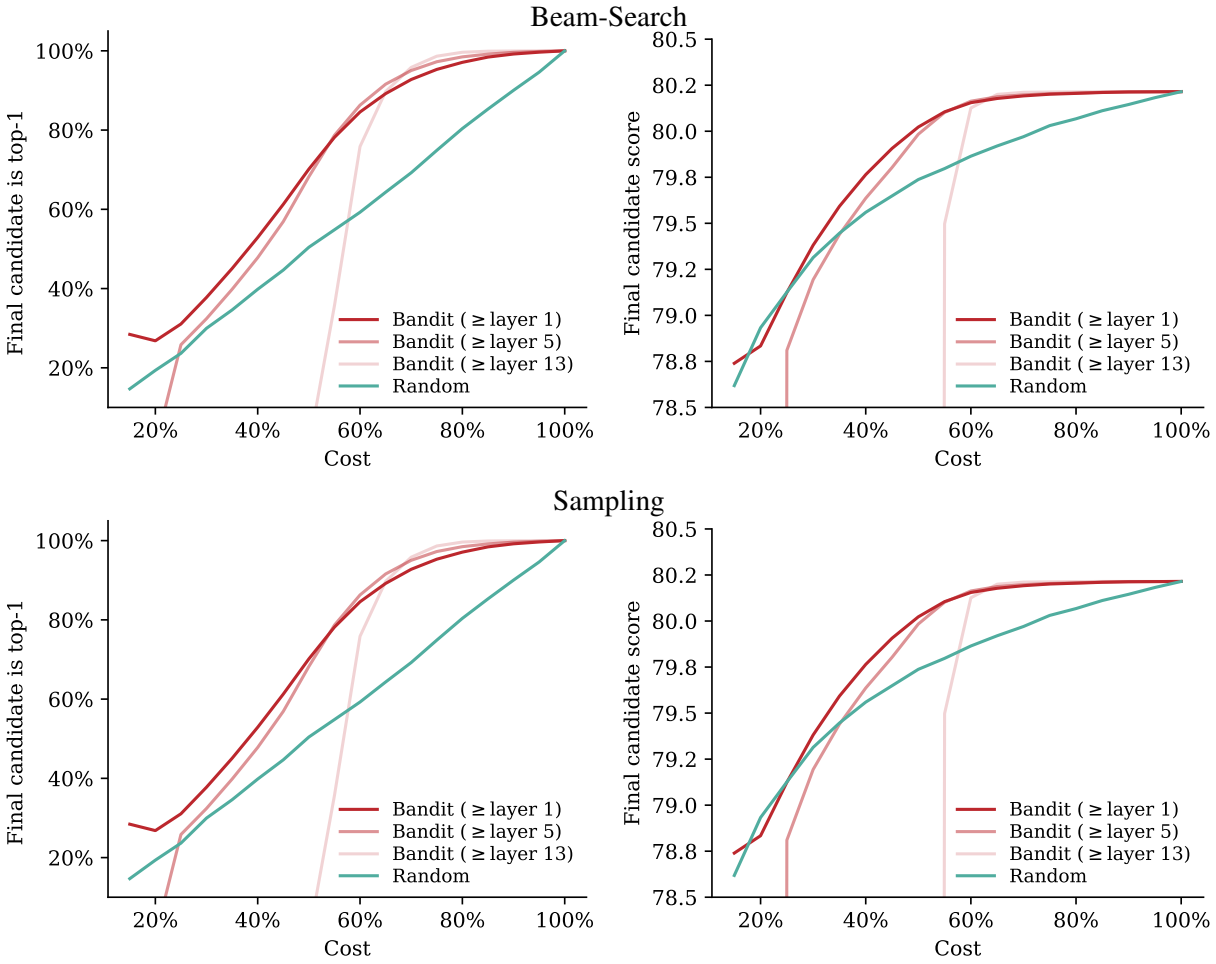

Figure 9: Ablation for the Upper Confidence Bound bandit by forcefully computing the first few layers. Quality is measured in terms of the average final candidate score and the proportion to top-1 candidates selected. We plot these measures for various evaluation budgets. Cost (or budget) is given relative to calculating the full COMET scores for all candidates (100%).

		15%	25%	35%	45%	55%	65%	75%	85%	95%
Beam-Search Final candidate is top-1	Bandit ($\gamma = 1.0$)	28.4%	31.1%	45.1%	61.3%	78.2%	89.2%	95.3%	98.4%	99.7%
	Random	14.7%	23.6%	34.6%	44.7%	54.8%	64.3%	74.9%	85.3%	94.6%
	LogProb Sum	17.1%	27.2%	36.9%	46.8%	56.6%	66.2%	76.2%	86.0%	95.7%
	LogProb Avg	13.7%	23.2%	32.5%	42.1%	51.9%	62.3%	72.4%	83.2%	94.4%
Beam-Search Final candidate score	Bandit ($\gamma = 1.0$)	78.74	79.12	79.59	79.91	80.11	80.18	80.20	80.21	80.21
	Random	78.62	79.13	79.45	79.65	79.80	79.92	80.03	80.11	80.18
	LogProb Sum	78.47	79.03	79.37	79.59	79.76	79.89	80.01	80.10	80.18
	LogProb Avg	78.34	78.91	79.23	79.47	79.65	79.80	79.93	80.05	80.16
Sampling Final candidate is top-1	Bandit ($\gamma = 1.0$)	32.1%	33.9%	44.3%	57.1%	71.7%	83.7%	92.7%	97.5%	99.5%
	Random	15.1%	25.0%	35.8%	44.6%	55.9%	63.9%	75.4%	84.0%	95.2%
	LogProb Sum	15.8%	25.6%	34.3%	43.4%	52.6%	62.4%	72.4%	83.4%	93.4%
	LogProb Avg	9.3%	17.6%	27.1%	36.5%	47.1%	57.9%	68.8%	79.5%	92.1%
Sampling Final candidate score	Bandit ($\gamma = 1.0$)	77.87	78.22	78.61	78.85	79.04	79.10	79.11	79.11	79.11
	Random	77.99	78.33	78.58	78.70	78.84	78.90	78.98	79.03	79.08
	LogProb Sum	77.59	78.08	78.36	78.54	78.70	78.83	78.92	79.00	79.07
	LogProb Avg	77.41	77.88	78.16	78.38	78.54	78.70	78.82	78.94	79.04

Table 4: Quality of the candidates returned by the Upper Confidence Bound bandit. Quality is measured in terms of the average final candidate score and the proportion to top-1 candidates selected. We plot these measures for various evaluation budgets. Cost (or budget) is given relative to calculating the full COMET scores for all candidates (100%). Visualized in Figure 5.

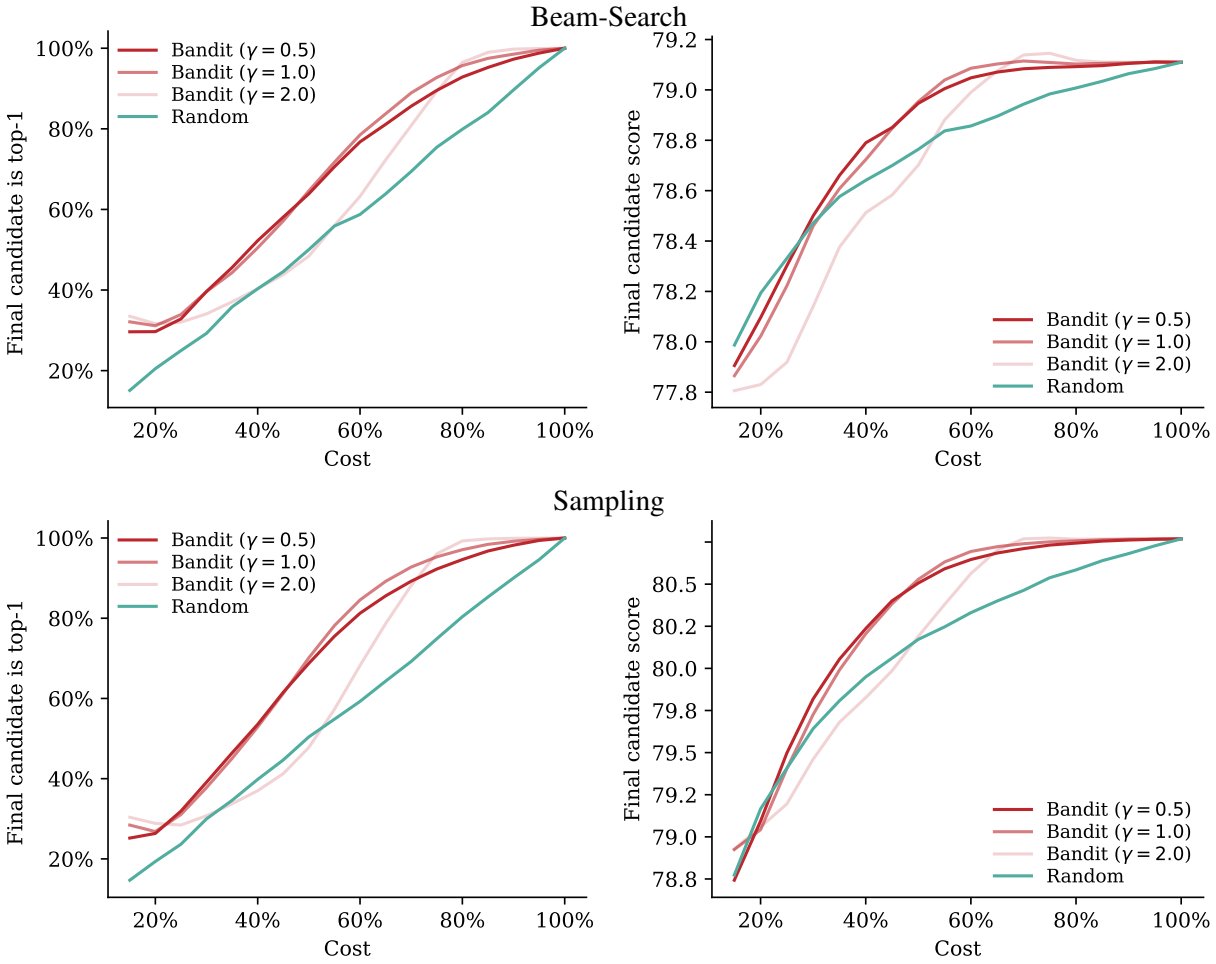

Figure 10: Ablation for the Upper Confidence Bound bandit with changing γ (exploitation-exploration trade-off). With higher γ, the algorithm explores even otherwise low-scoring candidates. Quality is measured in terms of the average final candidate score and the proportion to top-1 candidates selected. We plot these measures for various evaluation budgets. Cost (or budget) is given relative to calculating the full COMET scores for all candidates (100%).

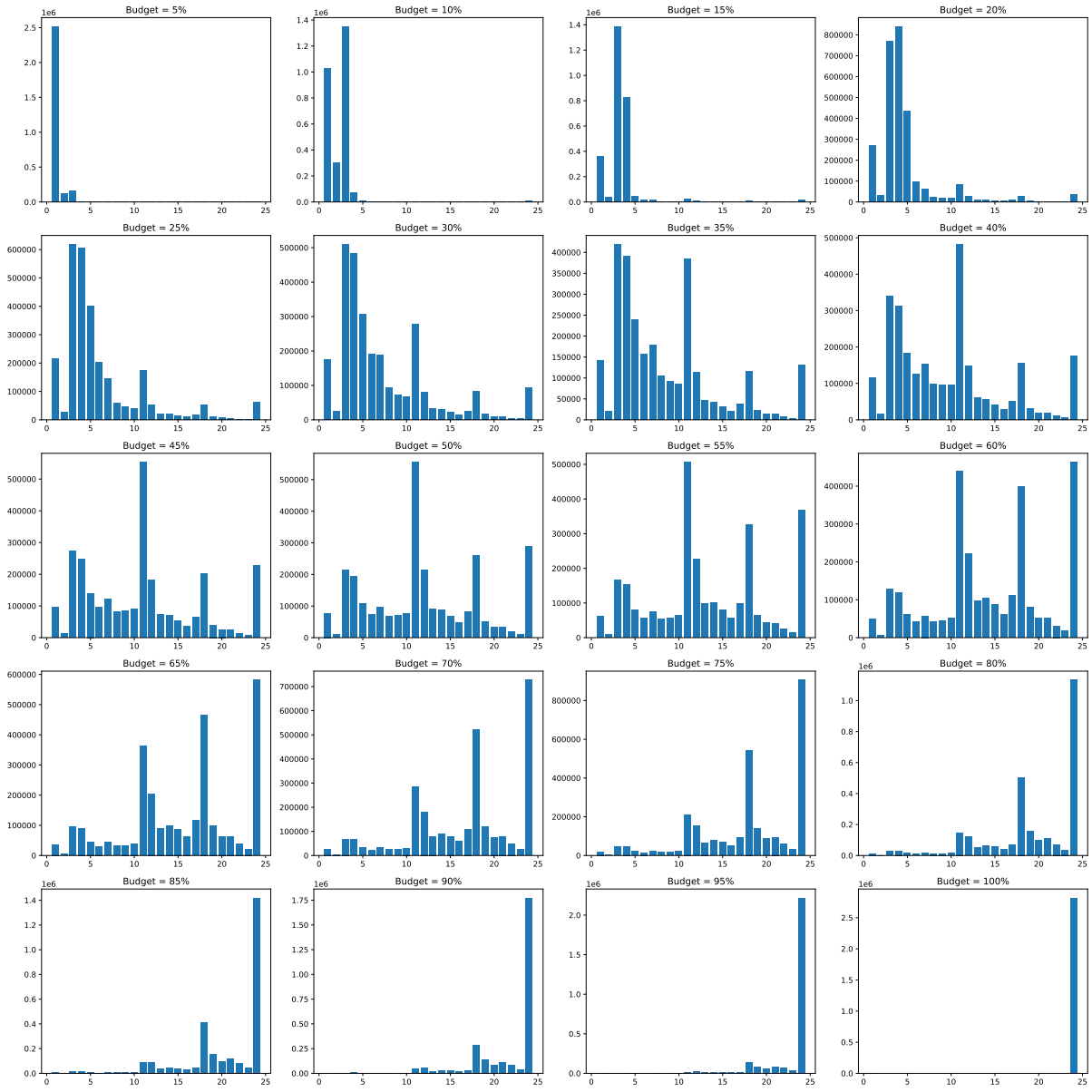

Figure 11: Distributions of the highest COMET layer scores calculated by UCB bandit (start layer = 1, $\gamma = 1.0$) across different budgets. The distributions are plotted for 5% budget increments, from 5% to 100%, on the test set with the candidates generated via sampling.

Layer \ Layers	01	03	05	07	09	11	13	15	17	19	21	23	24	Human
01	1.00	0.28	0.23	0.27	0.17	0.14	0.10	0.04	-0.01	-0.05	-0.07	-0.08	-0.06	-0.033
03	0.28	1.00	0.28	0.10	0.12	0.08	0.06	0.07	0.08	0.11	0.10	0.05	-0.01	-0.014
05	0.23	0.28	1.00	0.90	0.86	0.79	0.68	0.58	0.48	0.42	0.38	0.30	0.28	0.064
07	0.27	0.10	0.90	1.00	0.97	0.92	0.84	0.73	0.61	0.52	0.48	0.38	0.36	0.096
09	0.17	0.12	0.86	0.97	1.00	0.98	0.90	0.80	0.68	0.61	0.56	0.45	0.41	0.116
11	0.14	0.08	0.79	0.92	0.98	1.00	0.96	0.86	0.75	0.68	0.63	0.51	0.46	0.132
13	0.10	0.06	0.68	0.84	0.90	0.96	1.00	0.95	0.86	0.78	0.73	0.60	0.53	0.176
15	0.04	0.07	0.58	0.73	0.80	0.86	0.95	1.00	0.97	0.92	0.88	0.74	0.65	0.230
17	-0.01	0.08	0.48	0.61	0.68	0.75	0.86	0.97	1.00	0.98	0.96	0.83	0.73	0.264
19	-0.05	0.11	0.42	0.52	0.61	0.68	0.78	0.92	0.98	1.00	0.99	0.86	0.75	0.273
21	-0.07	0.10	0.38	0.48	0.56	0.63	0.73	0.88	0.96	0.99	1.00	0.91	0.80	0.283
23	-0.08	0.05	0.30	0.38	0.45	0.51	0.60	0.74	0.83	0.86	0.91	1.00	0.96	0.319
24	-0.06	-0.01	0.28	0.36	0.41	0.46	0.53	0.65	0.73	0.75	0.80	0.96	1.00	0.327

Baseline COMET

Layer \ Layers	01	03	05	07	09	11	13	15	17	19	21	23	24	Human
01	1.00	0.37	0.30	0.24	0.23	0.21	0.17	0.17	0.17	0.15	0.15	0.15	0.15	0.034
03	0.37	1.00	0.77	0.67	0.65	0.64	0.54	0.52	0.52	0.49	0.49	0.50	0.49	0.159
05	0.30	0.77	1.00	0.95	0.93	0.82	0.72	0.70	0.70	0.66	0.66	0.66	0.65	0.207
07	0.24	0.67	0.95	1.00	0.99	0.87	0.77	0.75	0.74	0.70	0.69	0.69	0.68	0.221
09	0.23	0.65	0.93	0.99	1.00	0.88	0.78	0.76	0.75	0.71	0.70	0.70	0.69	0.221
11	0.21	0.64	0.82	0.87	0.88	1.00	0.94	0.91	0.89	0.82	0.81	0.81	0.80	0.251
13	0.17	0.54	0.72	0.77	0.78	0.94	1.00	0.98	0.97	0.88	0.87	0.86	0.85	0.278
15	0.17	0.52	0.70	0.75	0.76	0.91	0.98	1.00	0.99	0.91	0.89	0.89	0.88	0.281
17	0.17	0.52	0.70	0.74	0.75	0.89	0.97	0.99	1.00	0.92	0.91	0.90	0.89	0.281
19	0.15	0.49	0.66	0.70	0.71	0.82	0.88	0.91	0.92	1.00	0.99	0.98	0.98	0.310
21	0.15	0.49	0.66	0.69	0.70	0.81	0.87	0.89	0.91	0.99	1.00	1.00	0.99	0.312
23	0.15	0.50	0.66	0.69	0.70	0.81	0.86	0.89	0.90	0.98	1.00	1.00	1.00	0.310
24	0.15	0.49	0.65	0.68	0.69	0.80	0.85	0.88	0.89	0.98	0.99	1.00	1.00	0.309

Early-Exit COMET

Layer \ Layers	01	03	05	07	09	11	13	15	17	19	21	23	24	Human
01	1.00	0.14	-0.17	-0.17	-0.14	-0.14	-0.14	-0.14	-0.15	-0.13	-0.12	-0.12	-0.12	-0.057
03	0.14	1.00	0.11	0.10	0.12	0.09	0.05	0.05	0.05	0.06	0.03	0.03	0.03	0.018
05	-0.17	0.11	1.00	0.86	0.78	0.71	0.66	0.65	0.64	0.58	0.56	0.56	0.56	0.186
07	-0.17	0.10	0.86	1.00	0.97	0.80	0.74	0.73	0.73	0.66	0.63	0.63	0.62	0.220
09	-0.14	0.12	0.78	0.97	1.00	0.84	0.77	0.76	0.76	0.68	0.65	0.65	0.65	0.231
11	-0.14	0.09	0.71	0.80	0.84	1.00	0.95	0.92	0.91	0.81	0.78	0.77	0.77	0.267
13	-0.14	0.05	0.66	0.74	0.77	0.95	1.00	0.99	0.98	0.87	0.83	0.82	0.82	0.284
15	-0.14	0.05	0.65	0.73	0.76	0.92	0.99	1.00	0.99	0.88	0.84	0.83	0.83	0.285
17	-0.15	0.05	0.64	0.73	0.76	0.91	0.98	0.99	1.00	0.89	0.85	0.84	0.84	0.287
19	-0.13	0.06	0.58	0.66	0.68	0.81	0.87	0.88	0.89	1.00	0.97	0.96	0.96	0.320
21	-0.12	0.03	0.56	0.63	0.65	0.78	0.83	0.84	0.85	0.97	1.00	1.00	0.99	0.324
23	-0.12	0.03	0.56	0.63	0.65	0.77	0.82	0.83	0.84	0.96	1.00	1.00	1.00	0.324
24	-0.12	0.03	0.56	0.62	0.65	0.77	0.82	0.83	0.84	0.96	0.99	1.00	1.00	0.325

Early-Exit COMET (separate heads)

Table 5: Pearson correlations between intermediate layer outputs (green) and between intermediate layer outputs and humans (purple) for unsupervised Early-Exit based on standard COMET, supervised Early-Exit with a single regression head, and supervised Early-Exit with separate regression head for each layer.

G Partial COMET

Most quality estimation models, such as COMET, struggle with partial generations and other non-standard inputs (Zouhar et al., 2024). This is also crucial in applications like speech translation, where the input is often segmented into smaller chunks (Sperber et al., 2024), potentially resulting in partial sentences or incomplete translations that current quality estimation models struggle to handle effectively (Amrhein and Haddow, 2022, Appendix D1).

In this section, we introduce *Partial COMET* (addressing non-standard input obstacle), ensuring reliable quality assessments for incomplete outputs. We show that also beyond evaluation, partial quality estimation can assess translation quality early in the translation generation process, and help discard unpromising candidates from beam search or sampling-based methods, allowing only the most promising candidates to be further expanded. (Appendix G)

Machine translation quality estimation systems provide an assessment of model output quality. However, in many cases, we wish to know the quality estimation even before a full translation is produced, for example in setups with unclear segmentations, such as simultaneous speech translation, or in the middle of the generation process. Using a quality estimation system trained on full translation candidates on partial generations will lead to lower scores because the translation is technically not correct. In this section, we propose a model that is able to score even partial generations, which are prefixes of the original translations. This can be then used during beam search or batch sampling to discard unpromising candidates.

G.1 Modeling

For partial generations, we do not yet know the full translation, but only its prefix: $t_{<i}$. Using this as an input to a quality estimation model that expects a full translation would result in a unfairly low scores. Therefore, we explicitly train a function for partial generations f_p, which sees partial generations during training:

$$\mathcal{L}_2(y, f(s, t_{<p})) \tag{9}$$

This way, the quality estimation model predicts final translation scores based on just the translation prefix. See Example 1.

Orig.	Hi there! Let's go eat, no? Hallo du! Lass uns essen gehen, oder?	→ 65
Part.	Hi there! Let's go eat, no? Hallo du! Lass—	→ 65

Example 1: Setup for partial translation quality estimation. The quality estimation model sees only half the translation but predicts the original human score for the whole translation.

To make this applicable for generation in MT models, we need to know when to stop in the middle of the translation. For this, we use a heuristic based on 25%, 50%, and 75% of the source segment length. To account for different language verbosities, we multiply the portion of the source length with fertility for each language pair, such as 1.1 for English→German. So for 50%, the model sees $t_{<1.1\times|s|\times 50\%}$. For each training segment we randomly choose whether 25%, 50%, 75%, or 100% is revealed.

G.2 Results

Full COMET falls short. We show the results, as measured by Pearson correlation across WMT23 in Table 6. The full COMET strongly underperforms on partial generations. Therefore, this quality estimation model is not suitable for evaluating incomplete segments during generation. In contrast, the partial quality estimation model correlates much more. This can also be due to human annotators overfocusing on the beginning of the translations when providing assessment scores (Magooda and Marcjan, 2020), or the quality estimation system picking up spurious correlations that give away the score (Zouhar et al., 2024).

Segments	Full COMET	Partial COMET
Partial 25%	0.097	0.210
Partial 50%	0.108	0.250
Partial 75%	0.155	0.283
Original 100%	0.327	0.318

Table 6: Pearson correlation in original and partial (half length) translation evaluation setups. The models are either trained on full or partial generations.

Partial COMET prunes generations. To show that the partial COMET is useful also in practice, we consider a generative model, such as for machine translation. In this case, the model produces multiple generations at the same time either using parallel sampling or beam-search. Higher beam

count or parallel samples generally lead to better performance, but also take much more compute. In this setup, partial COMET is useful because it can prune unpromising generations.

We use the the 600M-parameter distilled NLLB model (Team et al., 2022) with beam search and parallel sampling of 200[7]. However, instead of generating all 200 candidates until the end, we perform reranking with partial COMET when only 25% of the candidate's output has been generated. Then, we take only a fraction to continue in the generation of the main model. At 50%, we again perform the intermediate reranking, and once again at 75%.

We try all combinations of pruning bottom 0%, 25%, 50%, or 75% at each of 25%, 50%, and 75% target lengths. The results in Figure 12 show that by using Partial COMET, we are more likely to prune lower-quality candidates than with the original COMET, which itself performs better than random pruning.

Limitations. For Partial COMET, in some cases, the human score is not aligned with the translation substring, such as when the error which cause a lower human score is not present in the substring. This can be further remedied by word-level quality estimation, where considering only a substring to evaluation also automatically select only the present word-level errors.

G.3 Works on Segmentation-Robust QE

Learned automated metrics, such as COMET or MetricX (Juraska et al., 2024) inherit all problems of statistical machine learning, by expecting the input to come from a particular distribution (Zouhar et al., 2024). Amrhein and Haddow (2022) hint, and we confirm, that using a quality estimation model trained on full translations on incomplete segments leads to poor correlations. Justifiably so, because from the perspective of the quality estimation model, the translation is thus incorrect. This is problematic during decoding or in cases where the translation segmentation is unclear, such as for speech translation (Akhbardeh et al., 2021; Salesky et al., 2023; Han et al., 2024; Sperber et al., 2024).

[7]For computational tractability, we run this evaluation on a subset of WMT 2023 data, sampling 2000 examples for each language pair.

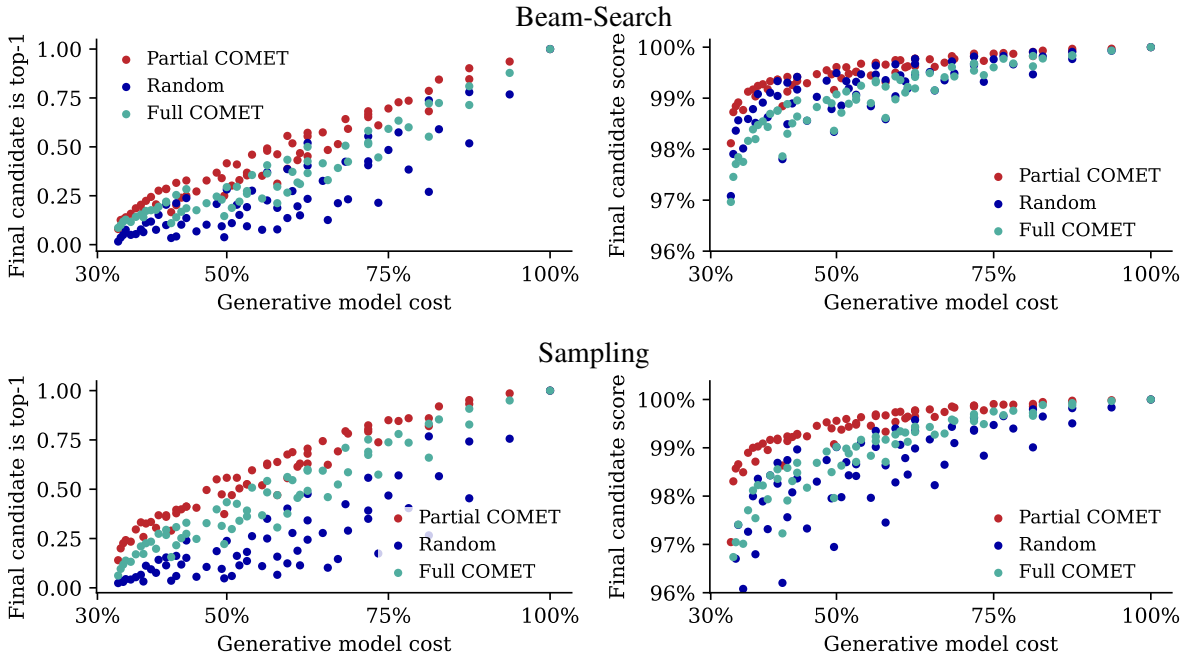

Figure 12: Proportion of the pruning process leading to the top candidate being chosen (left) or final candidate score (right) with respect to the computation cost of the generative model.