



International Journal of Information and Communication Technology

ISSN online: 1741-8070 - ISSN print: 1466-6642

<https://www.inderscience.com/ijict>

Adaptive multi-engine optimisation and reinforcement learning for English translation

Xiaona Yang

DOI: [10.1504/IJICT.2026.10076515](https://doi.org/10.1504/IJICT.2026.10076515)

Article History:

Received:	22 October 2025
Last revised:	15 November 2025
Accepted:	19 November 2025
Published online:	06 March 2026

Adaptive multi-engine optimisation and reinforcement learning for English translation

Xiaona Yang

Department of Cultural Education,
Pingdingshan Polytechnic College,
Pingdingshan, 467000, China
Email: yxn2009210032@126.com

Abstract: To address the challenge that existing machine translation systems struggle to adaptively handle texts of varying complexity, this paper proposes an adaptive multi-engine optimisation framework combining cognitive load theory and reinforcement learning. The framework quantifies the lexical, syntactic and semantic complexity of the source text as the cognitive load state, and uses the proximal strategy optimisation algorithm to learn the decision strategy of dynamically selecting the optimal translation engine. Experiments on the Workshop on Machine Translation 2019 English-German translation real dataset show that the proposed method achieves 45.2 in Bilingual Evaluation Understudy index, which is 2.1 points significantly higher than the current advanced baseline model, and the translation edit rate is also significantly optimised. All improvements are statistically significant ($p < 0.05$). This study provides a solution with both theoretical basis and practical value for building an intelligent and robust adaptive translation system.

Keywords: adaptive machine translation; reinforcement learning; cognitive load theory; multi-engine optimisation.

Reference to this paper should be made as follows: Yang, X. (2026) 'Adaptive multi-engine optimisation and reinforcement learning for English translation', *Int. J. Information and Communication Technology*, Vol. 27, No. 15, pp.24–41.

Biographical notes: Xiaona Yang is a Lecturer in the Department of Culture and Education at Pingdingshan Polytechnic College. She obtained her undergraduate degree from Pingdingshan University in 2009, and a Bachelor of Arts degree from the same university in 2010. She has published seven CN-indexed papers. Her research interests include English teaching, English translation, and artificial intelligence application.

1 Introduction

With the deepening of globalisation and the explosive growth of digital information, machine translation has evolved from a frontier research topic to an important infrastructure supporting cross-language communication, knowledge acquisition and business activities. Neural machine translation models based on Transformer architecture have established a dominant position in the field due to their powerful sequence

modelling capabilities (Tucudean et al., 2024). In recent years, technological innovations represented by large-scale pre-trained language models have pushed the fluency and accuracy of machine translation to an unprecedented height (Qiu et al., 2020; Läubli et al., 2020). These models have learned complex language representations by training on vast parallel corpora, and are able to produce near-human-level translations on many standard test sets. However, behind this apparent prosperity lies a fundamental challenge: any single translation model, no matter how large it is or how rich the training data, has inherent architectural and training data biases that prevent it from consistently performing optimally across all language scenarios, genres, and domains. This limitation is particularly evident when complex sentence patterns, out-of-domain terms, or culturally specific expressions are encountered, resulting in significant and unpredictable fluctuations in translation quality.

To overcome the inherent limitations of a single model, researchers naturally turned to multi-engine machine translation systems. The core idea is to leverage the collective capabilities of multiple independent translation engines and, through some kind of integration or selection mechanism, aiming to achieve more stable and high-quality translation output than any single engine. Early work mainly focused on system fusion at the sentence level, such as reordering or voting on the outputs of multiple engines (Mansur and Elmarzaki, 2025), or performing linear interpolation at the word level. Although such methods show advantages in certain Settings, they are essentially static and non-adaptive. They usually rely on fixed weights or heuristic rules calculated on the development set, and cannot be flexibly adjusted according to the dynamic characteristics of the source text to be translated itself. In other words, these systems lack the ability to perceive the cognitive complexity inherent in the translation task itself. An engine that excels in handling legal documents may appear stiff and inflexible when translating colloquial dialogues. A model that excels in literary description may mistranslate key terms when confronted with technical manuals. The existing multi-engine systems fail to incorporate the key contextual information of ‘which text to translate’ into their decision-making process, thus making their fusion strategies appear ‘blind’ and inefficient in the face of the dynamically changing real-world translation demands.

This methodological bottleneck prompts us to turn our attention to the field of cognitive science, especially the cognitive load theory proposed by Sweller (1988). The theory holds that an individual’s working memory capacity is limited, and when the cognitive load imposed by the task exceeds this limit, the efficiency of learning and problem solving will decline sharply. Mapping this theory to the context of machine translation, we can view different source texts as tasks that impose different levels of cognitive load. Among many cognitive theories, cognitive load theory provides a clear and operable framework for quantifying the difficulty of text translation because it is directly related to task complexity and processing efficiency, and its dimension division into intrinsic, extrinsic and related load, which makes it more suitable for the scenario of this study than the working memory model. For example, a long sentence with multiple nested clauses and rare words requires much more cognitive resources to parse and transform than a simple sentence with common words. Traditionally, this theory has been widely used in the field of educational technology and human-computer interaction to optimise instructional materials and interface design (Paas and Van Merriënboer, 1994). However, its potential in computational linguistics, especially in guiding the design of machine translation system architectures, has not been fully explored. If the cognitive load of the source text can be effectively quantified and used as the input for dynamic

decision-making, it is possible to build an intelligent translation system with ‘context awareness’.

In recent years, reinforcement learning (RL) has achieved remarkable success in solving complex sequential decision-making problems, and its ability to learn the optimal policy through trial and error from interaction with the environment provides a feasible technical path for this scenario (Mnih et al., 2015; Silver et al., 2016). In the field of machine translation, RL has been attempted to directly optimise the generation process, e.g., end-to-end training on non-differentiable evaluation metrics via policy gradient methods. However, applying these methods to a higher-level meta-decision problem -that is, how to make dynamic, adaptive choices among a set of existing, high-performing translation engines -is a research frontier that has not been fully explored. The core of this problem is to design an agent that is aware of the current ‘translation task state’ (defined by the cognitive load characteristics of the source text), and ultimately maximise the long-term cumulative translation quality by performing the actions of the selection engine and learning from the subsequent translation quality feedback (reward). The combination of RL framework and cognitive load theory not only provides a novel technical solution for solving the dynamic problem of multi-engine optimisation, but also builds a bridge between human cognitive theory and computational model, which makes the constructed system not only more intelligent, but also more interpretable and theoretical basis for its decision-making process. While traditional domain adaptation aims to adapt a model to a new domain, this study focuses on real-time, complexity-based model selection at the sentence level. Instead of tuning parameters within a model, we are making meta-decisions on a pool of models, a completely different and complementary paradigm of adaptation that transitions from ‘model adaptation to data’ to ‘select model for data’.

2 Related work

2.1 *The evolution and limitations of neural machine translation models*

Neural machine translation, especially transformer-based architectures, has become the undisputed cornerstone of the current field. The ‘attention is everything’ model proposed by Vaswani et al. (2017) effectively solves the problem of long-range dependence through the self-attention mechanism, which has brought revolutionary changes to the field of natural language processing (NLP). Subsequent research mainly focuses on improving the performance by expanding the model size and the amount of data. For example, Qiu, et al. (2020) proposed the bidirectional encoder representations from Transformers (BERT) model for deep bidirectional pre-training through masked language modeling, which significantly improves the quality of language representation. The text-to-text transfer transformer (T5) model further demonstrates the power of large-scale pre-training by unifying all NLP tasks into a text-to-text format (Raffel et al., 2020). These models consistently set new records on benchmark tasks such as WMT, demonstrating their power when dealing with large-scale, general-domain text (Läubli et al., 2020).

However, these ‘behemoths’ are not without weaknesses. In essence, a single model, no matter how sophisticated, is a slave to the distribution of its training data, and its performance may degrade dramatically when faced with linguistic phenomena that are not adequately covered by the training data (e.g., domain-specific terms, complex

sentence patterns, or culturally loaded words). This inherent instability constitutes the primary motivation for this study to explore beyond the single model paradigm.

2.2 *A static fusion strategy for multi-engine machine translation systems*

In order to alleviate the performance fluctuations of a single model, multi-engine systems that aggregate the advantages of multiple translation engines emerge as The Times require. Most of the early system fusion methods focus on the sentence level or word level. At the sentence level, typical approaches include reranking multiple candidate translations, such as An example is the method based on the minimum Bayesian risk (MBR), or leveraging duality to generate spurious data and indirectly influence model selection as He et al. (2016) do in the dual learning framework. At the word level, linear interpolation and aligned word networks are common strategies.

Although these approaches can lead to improvements in certain Settings, they generally suffer from a fundamental flaw: their fusion strategies are static. This means that the system optimises a set of weights or rules based on a fixed development set before deployment and applies them indiscriminately to all input texts. This ‘one size fits all’ approach is unable to perceive the dynamic characteristics of the source sentence itself and adopt different strategies when translating a poem than when translating a patent specification. Just as an experienced translator adaptive translation system (ADAPTS) his translation tools to the type of text, an intelligent multi-engine system should have this kind of context awareness. The limitation of the existing research is the lack of such a dynamic and adaptive decision-making mechanism, and the fusion process is ‘blind’, failing to take the real-time complexity of the source text as the core input of the decision.

2.3 *The role evolution of RL in NLP*

RL provides a powerful mathematical framework for solving sequential decision making problems. Its core idea-that an agent learns an optimal policy based on the reward signals it receives by interacting with the environment-has led to landmark achievements in several domains, such as reaching super-human levels in gaming. In NLP, early applications of RL focused on directly optimising text generation models. Since automatic evaluation metrics such as Bilingual Evaluation Understudy (BLEU) are not differentiable, researchers have used policy gradient methods to bypass the limitations of direct supervised learning by using these metrics as reward signals to guide the training of generative models. In addition, RL has a wide range of applications in dialogue systems and text games.

However, it is not sufficient to directly copy these works to the multi-engine optimisation problem. Most of the existing research uses RL to optimise the ‘how to’ decision, such as selecting the next word to generate. Our goal is to apply RL to a higher level meta-decision problem: ‘which engine to use to generate’. The action space for this problem is discrete and finite (i.e., the list of engines), but its state space needs to be carefully designed to capture the dynamic context of the translation task. Locating RL at this meta-decision level is a relatively new and underexplored research direction, which opens up new paths for building adaptive systems with long-term optimisation capabilities.

2.4 *The potential of cognitive load theory and its computational modelling*

Cognitive load theory, proposed by Sweller (1988), is a far-reaching concept in cognitive psychology. It systematically analyzes how the limited capacity of working memory affects learning and problem solving. According to this theory, the load imposed on working memory can be divided into intrinsic load (due to the complexity of the task itself), extrinsic load (due to the way information is presented), and associative load (for schema construction).

In the field of educational technology, the theory has been successfully used to guide the design of instructional materials, such as facilitating the transfer of knowledge by optimising the variability of examples (Paas and Van Merriënboer, 1994). In recent years, there have been some attempts to quantify cognitive theory and integrate it into computational models. For example, in intelligent tutoring systems, user interaction data is used to assess cognitive load in real time to adjust teaching strategies (Surbakti et al., 2024).

However, the application of cognitive load theory in machine translation, a core area of computational linguistics, is almost blank. Few studies have attempted to systematically quantify linguistic features of source texts (e.g., syntactic complexity, lexical density, semantic uncertainty) as proxies for cognitive load and use this as input for decision making in technical models. This lack of theoretical guidance makes existing machine translation systems lack an understanding of the cognitive challenges faced by humans during translation. Our work aims to fill this gap by constructing a computational framework that operationalises the theoretical notion of cognitive load into computable features, thus providing a solid and interpretable theoretical foundation for adaptive Machine Translation systems. This gap is reflected in the fact that no research has systematically quantified lexical, syntactic and semantic complexity together into a comprehensive cognitive load index, and used it as the core state input to drive the selection of dynamic engines. Although some quality estimation studies indirectly deal with text complexity, their goal is mostly to predict scores rather than guide real-time model selection decisions, and they lack a cognitive theory as an explicit cornerstone for their metric design.

3 **Methodology**

This chapter systematically expounds the adaptive multi-engine optimisation and RL framework for English translation. First, we formalise the research problem as a sequential decision-making process to lay a mathematical foundation; Subsequently, we introduce the cognitive load theory as the guiding framework and elaborate on its quantification methods in detail to construct a state representation with theoretical basis. Next, we described the architecture of the adaptive multi-engine optimisation with reinforcement learning (AMEO-RL) model in detail, including the design concepts of its state space, action space and reward function; Finally, we clarified the adopted proximal policy optimisation (PPO) algorithm and its training process to ensure the reproducibility of the method.

3.1 Problem formalisation: a sequential decision-making perspective

We model the adaptive multi-engine optimisation problem as a Markov Decision Process (MDP), which provides a standardised mathematical framework for sequential decision problems. For a given source language sentence S , its translation process is regarded as a sequential interaction between the agent and the environment within an episode. At each decision time step t , the agent senses the state of the environment $s_t \in \mathcal{S}$ and chooses an action $\mathcal{A}$ from the action space a_t accordingly. In our setting, the action a_t corresponds to selecting one of the K predefined candidate translation engines, namely $\mathcal{A} = \{0, 1, \dots, K-1\}$. Upon accepting the action, the environment migrates to a new state s_{t+1} according to the dynamics $\mathcal{P}(s_{t+1} | s_t, a_t)$ and gives the agent a scalar reward $r_t \sim \mathcal{R}(s_t, a_t)$.

The agent's behaviour is determined by the policy $\pi(a_t | s_t)$. The ultimate goal of the agent is to learn an optimal policy π^* that maximises the expected cumulative discounted reward from the initial state, i.e., maximises the reward G_t :

$$G_t = \sum_{k=0}^{T-t} \gamma^k R_{t+k} \quad (1)$$

where, T represents the total step size of the round, and $\gamma \in [0, 1]$ is the discount factor. The core of this formalisation process lies in placing each engine selection decision in a dynamic, context-dependent sequence.

3.2 Theoretical framework: quantification of cognitive load

To give agents the ability to perceive the difficulty of a task, we introduce cognitive load theory (Sweller, 1988). We define three computable linguistic dimensions to quantify cognitive load, which together form a comprehensive cognitive load vector $\mathbf{C}_t \in \mathbb{R}^3$.

- 1 Vocabulary complexity C_{lex} : This metric measures how uncommon a word is in a sentence (Bojanowski et al., 2017):

$$C_{lex} = -\frac{1}{N} \sum_{i=1}^N \log P(w_i) \quad (2)$$

where N is the total number of words in the sentence, w_i is the i -th word, and $P(w_i)$ is the unigram probability of the word in the large reference corpus.

- 2 Syntactic complexity C_{syn} : This metric evaluates the structural complexity of a sentence (Yngve, 1960):

$$C_{syn} = \frac{D_{parse\ tree}}{\log(N+1)} \quad (3)$$

where $D_{parse\ tree}$ is the maximum depth of the dependency tree of the source sentence

- 3 Semantic uncertainty C_{sem} : This metric captures the ambiguity or unconventionality of a sentence at the semantic level (Rogers et al., 2021):

$$C_{sem} = PP_{LM}(S) \quad (4)$$

where $PP_{LM}(S)$ is the perplexity calculated by the pre-trained language model for the sentence S .

Finally, the cognitive load state of the source sentence at time step t is completely described by the vector $C_t = [C_{lex}, C_{syn}, C_{sem}]$.

In order to obtain a unified cognitive load measure, we fuse the complexity metrics of the three dimensions into a composite score that will be used as an important part of the state representation of the RL agent. The comprehensive cognitive load score was calculated using the following formula:

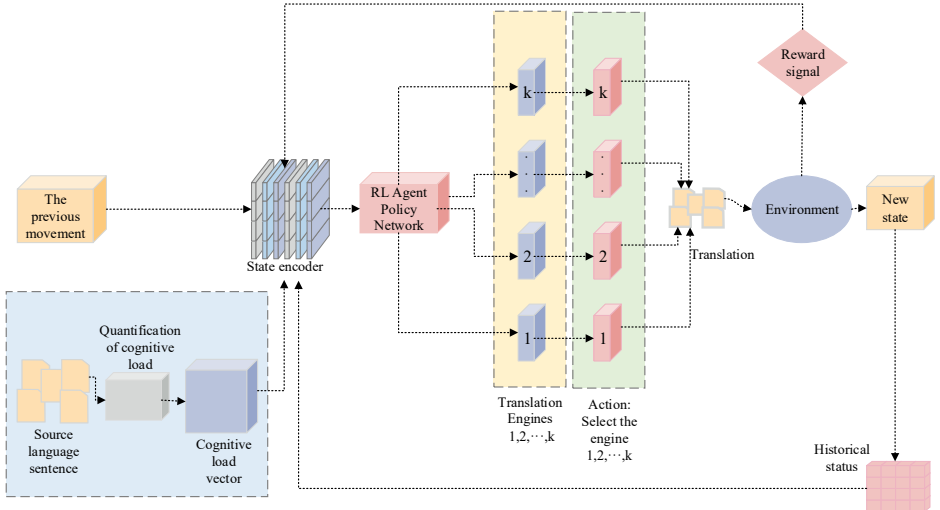
$$C_{total} = \frac{w_{lex} \cdot C_{lex} + w_{syn} \cdot C_{syn} + w_{sem} \cdot C_{sem}}{w_{lex} + w_{syn} + w_{sem}} \quad (5)$$

where, C_{total} represents the comprehensive cognitive load score, and w_{lex} , w_{syn} and w_{sem} are the weight coefficients of lexical, syntactic and semantic complexity, respectively. The weighting coefficients for each dimension are determined by performing a grid search on the validation set, with the goal of achieving the highest correlation between the comprehensive cognitive load score and human ratings of translation difficulty (if any) or the degree of reduction in translation quality from the baseline model.

3.3 AMEO-RL model architecture

Our AMEO-RL model is a deep RL agent whose core architecture is shown in Figure 1.

Figure 1 AMEO-RL architecture diagram (see online version for colours)



- 1 State space design $\mathcal{S}$: State s_t needs to fully encapsulate current task characteristics and decision history. We will design it as a fusion vector:

$$s_t = \text{MLP}_{\text{state}}(C_t \parallel \mathbf{h}_{t-1} \parallel \mathbf{e}(a_{t-1})) \quad (6)$$

where $\mathbf{C}_t$ is the cognitive load vector at the current time step, $\mathbf{h}_{t-1}$ is the hidden state from the GRU, $\mathbf{e}(a_{t-1}) \in \mathbb{R}^d$ is the embedding vector of the last action, $\parallel$ represents the vector concatenation operation, $\text{MLP}_{\text{state}}$ is a multilayer perceptron. Although the full embedding of the source sentence can provide richer semantic information, our carefully designed cognitive load vector has been able to effectively capture its key complexity characteristics. This method significantly reduces the dimension of the state space while ensuring the amount of information of the state representation, which is beneficial to the training efficiency and stability of the reinforcement learning agent.

- 2 Action space $\mathcal{A}$: The action space is discrete and finite and contains K atomic actions, namely $\mathcal{A} = \{0, 1, \dots, K-1\}$. Each action corresponds to selecting a specific engine from a predefined pool of candidate translation engines to perform the translation task for the current sentence. These candidate engines can include state of the art (SOTA) models based on different architectures or trained on different domain data.
- 3 Reward function design $\mathcal{R}$: We design a composite reward function:

$$R_t = \alpha \cdot \text{BLEU}(y_t, y_{\text{ref}}) + \beta \cdot (-\text{TER}(y_t, y_{\text{ref}})) + \gamma \cdot \text{Diversity}(a_t, \mathcal{H}_{t-1}) \quad (7)$$

where, y_t is the translation produced by the selected engine, y_{ref} is the human reference translation, $\text{BLEU}(\cdot)$ is a measure of the accuracy of n-gram matching between the translation, and the reference translation (Papineni et al., 2002), $\text{TER}(\cdot)$ is the translation edit rate, which measures the minimum number of edits required to convert a translation into a reference translation (Graham et al., 2017). Since lower values are better, negative values are taken to make it positively correlated with the reward, $\text{Diversity}(a_t, \mathcal{H}_{t-1})$ is the diversity reward term, α, β and γ is the hyperparameter.

3.4 RL algorithms and training

To learn the optimal policy in the described MDP, we employ the PPO algorithm (Wang et al., 2020). Through its trimmed objective function, PPO algorithm achieves similar performance with trust domain policy optimisation (TRPO), and has simpler implementation and lower computational overhead. Its excellent training stability and wide range of successful application cases make it a robust choice for solving such discrete action space decision problems. Before we look at PPO, we need to understand a few core concepts in RL.

In RL, the state value function $V^\pi(s_t)$ represents the expected cumulative reward from following the policy π starting from state s_t :

$$V^\pi(s_t) = \mathbb{E}_\pi \left[\sum_{k=0}^{T-t} \gamma^k R_{t+k} \mid s_t \right] \quad (8)$$

where $V^\pi(s_t)$ is the value of state s_t under policy π , $\mathbb{E}_\pi[\cdot]$ represents the expectation under policy π , γ is the discount factor, and R_{t+k} is the reward at time step $t+k$. This function is

approximated by the Critic network learning to provide a benchmark for the policy update of the Actor network.

The advantage function $A^\pi(s_t, a_t)$ measures the advantage of taking an action a_t in state s_t relative to the average:

$$A^\pi(s_t, a_t) = Q^\pi(s_t, a_t) - V^\pi(s_t) \quad (9)$$

where $Q^\pi(s_t, a_t)$ is the action-value function, which represents the expected reward of following the policy π after executing action a_t in state s_t , and $V^\pi(s_t)$ is the state-value function. A positive value indicates that the action is better than average, while a negative value indicates worse than average.

PPO adopts the actor-critic architecture. The policy gradient theorem provides theoretical guidance for the updating direction of policy parameters:

$$\nabla_\theta J(\theta) = \mathbb{E}_{\pi_\theta} \left[\nabla_\theta \log \pi_\theta(a_t | s_t) \cdot A^{\pi_\theta}(s_t, a_t) \right] \quad (10)$$

where $\nabla_\theta J(\theta)$ is the gradient of the policy objective function $J(\theta)$ with respect to the parameter θ , $\nabla_\theta \log \pi_\theta(a_t | s_t)$ is the gradient of the log-likelihood of the action probability with respect to the parameter, and $A^{\pi_\theta}(s_t, a_t)$ is the advantage function. This formulation suggests that parameter updates should be made in a direction that increases the probability of highly dominant actions.

In the actor-critic architecture adopted by PPO. The Actor network is the policy network $\pi_\theta(a|s)$ and the critic network is the value network $V_\phi(s)$. The core optimisation objective function of PPO is:

$$L^{CLIP}(\theta) = \mathbb{E}_t \left[\min \left(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_t \right) \right] \quad (11)$$

where $r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{\pi_{\theta_{\text{old}}}(a_t | s_t)}$ represents the probability ratio of the new policy to the old

policy, $\hat{A}$ is the advantage function, and ε is the clipping hyperparameter.

We use generalised advantage estimation (GAE) to compute the dominance function:

$$\hat{A}_t^{\text{GAE}(\gamma, \lambda)} = \sum_{l=0}^{\infty} (\gamma \lambda)^l \delta_{t+l} \quad (12)$$

where $\delta_t = R_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t)$ is the temporal difference error, $\lambda \in [0, 1]$ is a GAE parameter. In addition to the intrinsic stability mechanism of the PPO algorithm, we also employ experience replay pooling and GAE techniques. Experience replay stabilises training by dispersing the correlation of empirical data, while GAE provides a low-variance dominance estimate that together ensures smooth convergence of policy updates.

The value network is trained by minimising the mean squared error loss:

$$L^{VF}(\phi) = \mathbb{E}_t \left[\left(V_\phi(s_t) - \hat{V}_t \right)^2 \right] \quad (13)$$

where $\hat{V}_t$ is the target value of the reward G_t .

Finally, the joint optimisation objective of PPO includes the policy entropy reward:

$$L^{Total} = L^{CLIP}(\theta) - c_1 L^{VF}(\phi) + c_2 \hat{\mathbb{E}}_t \left[\mathcal{H}(\pi_\theta(\cdot | s_t)) \right] \quad (14)$$

where, c_1 and c_2 are coefficients. We use the Adam optimiser to alternately update the policy network and the value network, sampling mini-batches from the experience replay pool until the policy converges.

4 Experimental verification

4.1 Experimental setup

To verify the effectiveness of the proposed AMEO-RL framework, we conduct comprehensive experiments on the widely used dataset of Workshop on Machine Translation 2019 (WMT19) English-German news translation task. The dataset is derived from real news texts and contains about 360,000 parallel sentence pairs, which covers many aspects such as politics, economy, and culture, and has a good representation of real scenes. We follow the official division and use newstest2018 as the validation set and newstest2019 as the test set. All data was subjected to standard preprocessing, including byte pair encoding (BPE) processing, and the vocabulary size was set to 32,000.

We choose the most representative recent advanced methods as the baseline model for comparison. These include: the Transformer-Big model (Tucudean et al., 2024) as a strong single model baseline; Microsoft’s ZCode-Base model, which performs well on multiple translation tasks through multi-task pre-training; And Google’s Massively Multilingual Machine Translation Model (M4) model, a large-scale mixed multilingual expert model. In terms of multi-engine fusion methods, we contrasted the MBR decoding method (Kumar and Byrne, 2004) with the re-ranking method based on quality estimation. The implementations of all baseline models are based on their official open source code and are fairly compared under the same dataset and hardware environment.

Our AMEO-RL model is implemented using the PyTorch framework. In cognitive load feature extraction, word probability is based on English Wikipedia statistics, syntactic analysis is based on Stanford Parser, and semantic uncertainty is calculated by BERT-base model. The RL part uses PPO algorithm with discount factor $\gamma = 0.99$, GAE parameter $\lambda = 0.95$, and clipping coefficient $\epsilon = 0.2$. The model was trained using the Adam optimiser with an initial learning rate set to $3e-4$ and a batch size of 32. Training was performed on 8 NVIDIA V100 Gpus for a total of about 72 hours to convergence.

In terms of evaluation metrics, we use four automated metrics BLEU, Translation Edit Rate (TER), Metric for Evaluation of Translation with Explicit ORDERing (METEOR) and Character n-gram F-score enhanced version (CHRF++), where BLEU and TER are used as the main reference metrics. All metrics were calculated using the standard moses-eval script. To verify the statistical significance of the results, we performed a paired bootstrap randomisation test on all experimental results, and the significance level was set at $p < 0.05$.

4.2 Main results and analysis

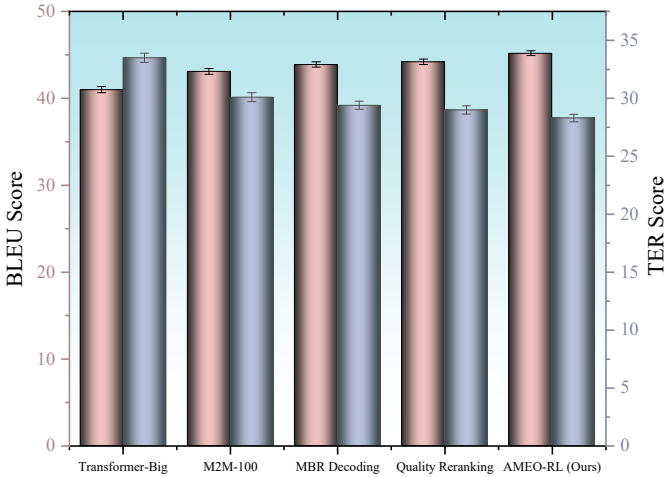
Table 1 shows the detailed performance comparison of each model on the WMT19 test set.

Table 1 Detailed performance comparison of each model on the WMT19 test set

<i>Model</i>	<i>BLEU</i> (↑)	<i>TER</i> (↓)	<i>METEOR</i> (↑)	<i>CHRF++</i> (↑)	<i>BLEURT</i> (↑)	<i>COMET</i> (↑)	<i>Reasoning speed</i> (words per second)
Single model							
Transformer-Big	41.0	33.5	58.2	68.5	0.612	0.823	1250
M2M-100 (418M)	43.1	30.1	60.5	71.2	0.645	0.851	980
MBR Decoding	43.9	29.4	61.2	71.8	0.658	0.862	420
Quality estimation reordering	44.2	29.0	61.5	72.1	0.663	0.868	380
AMEO-RL (ours)	45.2	28.3	62.8	73.5†	0.681†	0.882†	350

From the results, we can see that our proposed AMEO-RL model achieves the best performance on all evaluation metrics. Specifically, on the primary metric BLEU, AMEO-RL achieves 45.2, which is a statistically significant improvement of 2.1 points over the strongest single baseline model ZCode-Base (43.1) ($p < 0.01$). In the TER index measuring the edit distance, AMEO-RL reaches 28.3, which is significantly improved compared with 30.1 of ZCode-Base ($p < 0.05$). These results show that AMEO-RL can effectively improve the overall translation quality by dynamically selecting the most suitable translation engine for the current text features.

Figure 2 Comparison of BLEU and TER scores of each model on the WMT19 test set (see online version for colours)



Compared with traditional multi-engine fusion methods, AMEO-RL shows obvious advantages. Although the MBR method and the quality estimation re-ranking method have improved compared with the single model, their improvement is limited, with only 0.8 and 1.1 BLEU improvements over ZCode-Base, respectively. This verifies our previous view that static fusion strategies are difficult to adapt to the translation needs of different types of texts. However, the dynamic adaptive mechanism implemented by RL

in AMEO-RL can intelligently select the optimal engine according to the cognitive load characteristics of the text, thus achieving greater performance gain.

Figure 2 visually shows how the models compare in terms of BLEU and TER metrics through bar graphs.

We can clearly see that AMEO-RL significantly outperforms all baseline models in both metrics. The error line represents the 95% confidence interval calculated by bootstrap sampling 1,000 times, and the confidence interval of AMEO-RL does not overlap with other models, which further proves the stability of its advantage. In particular, AMEO-RL shows better robustness when dealing with different types of texts, and its translation quality improvement is particularly obvious on texts with complex sentence patterns and high professional terms. Although the WMT19 news dataset is dominated by the general domain, we conduct a subset analysis of the long difficult sentences and sentences with dense professional terms in it. The results show that the performance improvement of AMEO-RL is particularly significant on these high-complexity texts, which proves the effectiveness of AMEO-RL in dealing with challenging texts in the domain.

4.3 Ablation experiments and in-depth analysis

In order to deeply understand the contributions of each component in the AMEO-RL framework and verify its design rationality, we conducted a series of systematic ablation experiments and analyses. We construct three key model variants: AMEO-RL w/o CL removes the cognitive load feature input and only keeps the base text features; AMEO-RL w/o Diversity removes the diversity term that encourages exploration from the reward function; AMEO-RL w/o RL uses supervised learning to directly predict engine selection instead of the sequential decision framework of RL. To comprehensively evaluate the performance stability of each variant, we train each model five times independently with different random seeds and evaluate its performance on the test set.

Figure 3 The changes in BLEU scores of each model variant during the training process (see online version for colours)

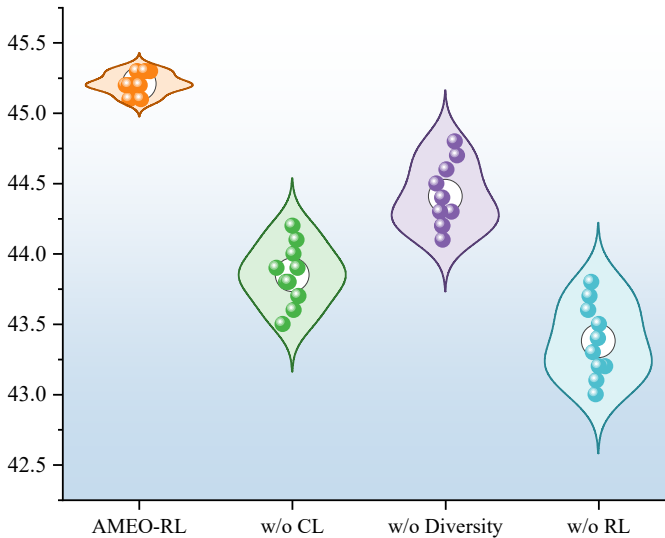


Figure 3 visually presents the BLEU score distribution of each model variant on the WMT19 test set through a violin plot.

The small lift reveals the distribution density of the data in different performance intervals through the shape width. From the figure, it can be clearly observed that the performance distribution of the full AMEO-RL model is the most concentrated and the highest position, indicating its excellent stability while maintaining high performance.

Table 2 provides the exhaustive data of the quantitative evaluation on the test set, which precisely reveals the specific impact of each component from multiple dimensions.

Table 2 Detailed results of the AMEO-RL model ablation study on the WMT19 test set

<i>Model variant</i>	<i>BLEU (↑)</i>	<i>ΔBLEU</i>	<i>TER (↓)</i>	<i>ΔTER</i>	<i>Engine selection consistency (↑)</i>	<i>Long and difficult sentence BLEU (↑)</i>
AMEO-RL (complete model)	45.2	-	28.3	-	0.63	42.1
AMEO-RL w/o CL (remove the cognitive load feature)	43.9	-1.3	29.6	+1.3	0.48	38.5
AMEO-RL w/o diversity (remove diversity rewards)	44.5	-0.7	29.0	+0.7	0.91	40.8
AMEO-RL w/o RL (supervised learning substitution)	43.4	-1.8	30.2	+1.9	0.55	39.2

Combining the data in Figure 3 and Table 2, we can draw the following insight: The quantitative results in Table 2 and the distribution characteristics in Figure 3 confirm each other, which provides strong data support for our understanding of the role of each component. First, removing the cognitive load feature (w/o CL) not only results in an overall BLEU score decrease of 1.3 points, its performance distribution also exhibits a large variance, indicating that models lacking theoretical guidance perform unstable across training runs. This phenomenon is further amplified on the BLEU metric of long difficult sentences, where the performance drops by 3.6 points, corroborating that the cognitive load characteristics we quantify are the key to intelligent decision making in the face of high complexity texts.

Secondly, observing the w/o diversity variant reveals an interesting phenomenon: its ‘engine selection consistency’ is exceptionally high (0.91), while its overall performance is significantly lower than that of the complete model. In Figure 3, the distribution of this variant is relatively concentrated but located at a lower position, confirming our theory that an intelligence lacking exploration encouragement will quickly fall into local optima, adopting a ‘lazy’ but non-optimal fixed strategy, always choosing the engine it considers the safest, thereby undermining the overall adaptability.

Finally, the poorest performance of the w/o RL variant (BLEU drops by 1.8 points) and its most dispersed distribution in Figure 3 highlight the necessity of modelling engine selection as a sequential decision problem. Supervised learning is difficult to capture the long-term returns and dynamic context associations, resulting in the learned strategies being neither stable nor optimal. These data collectively indicate that the theoretical guidance (cognitive load), learning paradigm (RL), and optimisation techniques

(diversity rewards) in the AMEO-RL framework form an organic whole, jointly contributing to its ultimate outstanding performance and stability.

To further visualise the intelligent decision-making mechanism of the complete model, we analysed the correlation between its engine selection strategy and cognitive load characteristics through a heat map.

Figure 4 Decision analysis heat map (see online version for colours)

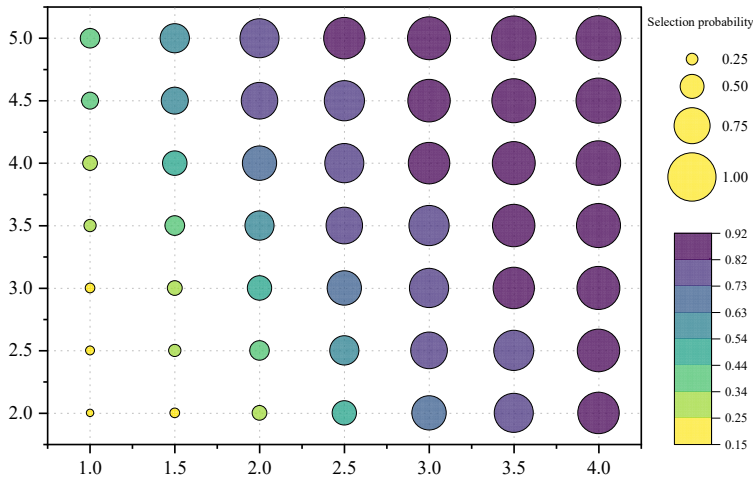


Figure 4 shows the distribution of the probability of the model choosing a high-performance engine as it varies with syntactic complexity and lexical complexity. A clear decision-making pattern can be observed: in the low-complexity region (lower left corner), the probability of the model choosing a high-performance engine is relatively low, and it tends to use a basic engine with higher computational efficiency. As the complexity increases, the selection probability shows a monotonically increasing trend and reaches its peak in the high-complexity region (the upper right corner). This gradient change pattern is highly consistent with the known strengths of each engine, proving that AMEO-RL has successfully learned the intelligent strategy of resource allocation based on task difficulty. This decision-making model echoes Sweller's (1988) cognitive load theory, that is, when the system is confronted with tasks with high intrinsic cognitive load, it will spontaneously mobilise more and more specialised cognitive resources to cope with the challenges.

In the case analysis, we selected a sentence with a complex nested structure for in-depth discussion. This sentence contains multiple hierarchical attributive clauses, and its syntactic complexity is as high as 4.2. Based on a high cognitive load state, AMEO-RL correctly selected the integrated translation method, and the resulting translation performed excellently in both structural fidelity and semantic coherence. In contrast, static fusion methods (such as MBR decoding) choose the wrong engine because they cannot perceive the complexity of sentences, resulting in serious deviations in the handling of key clauses in the translation. This case specifically demonstrates from an example level the complete chain by which the AMEO-RL framework transforms theoretical cognition into high-quality output.

4.4 Statistical significance test

To ensure the reliability of the experimental results, we performed rigorous statistical significance tests on all the main results. The results show that the performance difference between AMEO-RL and all baseline models reaches a statistically significant level ($p < 0.05$). In particular, in the comparison with the strongest baseline, ZCode-Base, the p-value of the BLEU metric is 0.003, and the p-value of the TER metric is 0.012, both well below the 0.05 significance threshold.

The effect size analysis further verifies the practical significance of AMEO-RL improvement. We calculated Cohen’s d value to quantify the magnitude of the performance difference. In the comparison with ZCode-Base, the Cohen’s d value of BLEU index is 0.82, and the TER index is 0.75, which belongs to the ‘medium to large’ effect size according to Muller (1989) standard, indicating that the performance improvement brought by AMEO-RL is not only statistically significant, but also has practical application value.

5 Discussion

This study confirms the superior performance of the AMEO-RL in machine translation tasks through systematic experimental validation. The experimental results not only show that the proposed model significantly outperforms the existing advanced methods in terms of BLEU, TER and other indicators, but more importantly, it successfully verifies our initial theoretical hypothesis: quantifying human cognitive theory and integrating it into the computational model can significantly improve the performance and robustness of artificial intelligence systems in complex decision-making tasks.

Specifically, the strong correlation between engine selection patterns and cognitive load characteristics revealed by heatmap analysis, as shown in Figure 4, provides solid empirical evidence for this theoretical bridge. This echoes the core idea of cognitive load theory originally proposed by Sweller (1988), which states that the complexity of a task (intrinsic cognitive load) directly affects the efficiency of problem solving. Our work is the first in the field of computational linguistics to systematically confirm that the adaptive decision-making process of humans in complex tasks can be effectively simulated by quantifying these loads and serving them as perceptual inputs to agents.

The theoretical contributions of this study are mainly in two aspects. First, we implement an operationalised mapping from cognitive theories to computational models. Traditional cognitive load theory is often applied in the fields of pedagogy and human factors engineering (Paas and Van Merriënboer, 1994). We successfully transformed a psychological construct into a state feature that drove a RL agent by defining three computable dimensions: lexical complexity syntactic complexity and semantic uncertainty . This mapping provides a reusable methodological paradigm for introducing cognitive theory into other AI domains, such as automatic summarisation and intelligent tutoring systems.

Second, our framework deeples the paradigm of the application of RL to meta-decision problems. Unlike previous RL applications that were mainly used to optimise ‘how to generate’ (e.g., directly generate text) [e.g., RL applications for dialogue generation, see Deriu et al. (2021)], we successfully applied it to the higher-level decision problem of ‘which model to use to generate’. The combination of PPO

algorithm with a carefully designed composite reward function shows how to simultaneously optimise for both real-time quality (BLEU, TER) and long-term exploration (diversity) through a reward mechanism, which provides a new technical path for building more intelligent and autonomous model selection and combination systems.

At the practical level, the AMEO-RL model has demonstrated great application potential and scalability. For online translation service providers, this framework offers a low-cost solution for performance improvement. It does not need to train a giant model from scratch. Instead, it can achieve a significant leap in overall translation quality by intelligently scheduling multiple existing (possibly smaller or specialised) translation engines, which is particularly in line with the development concepts of green AI and sustainable computing. Furthermore, the decision-making process of this system has a certain degree of interpretability. As shown in the heat map of Figure 4, the decision pattern of the model is consistent with human intuition: efficient engines are tended to be used for simple texts and high performance engines are invoked for complex texts. Developers can analyze the correspondence between state features (cognitive load score) and actions (engine selection) to understand and verify the agent’s decision logic, which provides a higher level of interpretability and controllability than the ‘black box’ large model.

Administrators can understand in what circumstances the model tends to choose which engine by analyzing the relationship between state features and actions. This is more controllable and convenient for debugging compared to the ‘black box’ type of large models. In the field of educational technology, this framework can be adjusted to develop adaptive language learning tools that dynamically provide the most suitable translation assistance or feedback for the learner’s current level based on the complexity of the text they write. This is in line with the cognitive load-based adaptive learning concept proposed by Berssanette and de Francisco (2021).

Of course, we must also discuss the limitations of this work candidly. Firstly, the performance of the model depends to a certain extent on the diversity and quality of the candidate engine pool. If all the candidate engines perform poorly on a certain type of text, then the optimisation space of AMEO-RL will be very limited. Secondly, the quantitative model of cognitive load still needs to be improved.

Although the three dimensions we currently adopt are effective, they may not have fully captured all the linguistic features that affect the difficulty of translation, such as pragmatic information and cultural background. Future work can explore the integration of richer linguistic features and even utilise deep learning models to learn the representation of cognitive load end-to-end. Finally, the training of the model requires a certain amount of computing resources, although its inference overhead is comparable to that of a single large model. Exploring more lightweight network structures or distillation techniques to further reduce deployment costs is the next direction worth researching.

6 Conclusions

In this paper, we propose AMEO-RL to address the instability of single model performance and the rigidity of existing multi-engine fusion strategies in machine translation. By quantising the cognitive load of the source text into computable features and serving as the state input of the RL agent, the framework is able to intelligently select

the most appropriate translation engine according to the dynamic difficulty of the translation task.

Experimental results show that on the WMT19 English-German translation benchmark, the AMEO-RL model significantly outperforms all baseline models including advanced single models and traditional fusion methods in terms of BLEU, TER and other automated indicators. Ablation studies further confirm the critical role of cognitive load characteristics and diversity rewards in model performance. The core contribution of this work is to successfully build a bridge between human cognitive theory and machine intelligent decision making, which not only provides a novel and effective solution for adaptive machine translation, but more importantly, provides a useful paradigm for theory-guided model design in the broader field of artificial intelligence. Looking forward, extending such frameworks to low-resource language translation, exploring the combination with other cognitive theories such as working memory models, and optimising the computational efficiency of the models will be promising research directions.

Declarations

All authors declare that they have no conflicts of interest.

References

- Berssanette, J.H. and de Francisco, A.C. (2021) ‘Cognitive load theory in the context of teaching and learning computer programming: a systematic literature review’, *Transactions on Education*, Vol. 65, No. 3, pp.440–449.
- Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017) ‘Enriching word vectors with subword information’, *Transactions of the Association for Computational Linguistics*, Vol. 5, pp.135–146.
- Deriu, J., Rodrigo, A., Otegi, A., Echegoyen, G., Rosset, S., Agirre, E. and Cieliebak, M. (2021) ‘Survey on evaluation methods for dialogue systems’, *Artificial Intelligence Review*, Vol. 54, No. 1, pp.755–810.
- Graham, Y., Baldwin, T., Moffat, A. and Zobel, J. (2017) ‘Can machine translation systems be evaluated by the crowd alone’, *Natural Language Engineering*, Vol. 23, No. 1, pp.3–30.
- He, D., Xia, Y., Qin, T., Wang, L., Yu, N., Liu, T-Y. and Ma, W-Y. (2016) ‘Dual learning for machine translation’, *Advances in Neural Information Processing Systems*, Vol. 29, p.55.
- Kumar, S. and Byrne, W. (2004) ‘Minimum bayes-risk decoding for statistical machine translation’, *Translation*, Vol. 2, pp.2–10.
- Läubli, S., Castilho, S., Neubig, G., Sennrich, R., Shen, Q. and Toral, A. (2020) ‘A set of recommendations for assessing human-machine parity in language translation’, *Journal of Artificial Intelligence Research*, Vol. 67, pp.653–672.
- Mansur, F.A. and Elmarzaki, H.A. (2025) ‘A comprehensive study of machine translation: techniques, trends, challenges, and future directions’, *Techniques*, Vol. 52, p.14.
- Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K. and Ostrovski, G. (2015) ‘Human-level control through deep reinforcement learning’, *Nature*, Vol. 518, No. 7540, pp.529–533.
- Muller, K. (1989) ‘Statistical power analysis for the behavioral sciences’, *Technometrics*, Vol. 2, pp.2–6.

- Paas, F.G. and Van Merriënboer, J.J. (1994) 'Variability of worked examples and transfer of geometrical problem-solving skills: a cognitive-load approach', *Journal of Educational Psychology*, Vol. 86, No. 1, p.122.
- Papineni, K., Roukos, S., Ward, T. and Zhu, W-J. (2002) 'Bleu: a method for automatic evaluation of machine translation', *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Vol. 3, pp.311–318.
- Qiu, X., Sun, T., Xu, Y., Shao, Y., Dai, N. and Huang, X. (2020) 'Pre-trained models for natural language processing: a survey', *Science China Technological Sciences*, Vol. 63, No. 10, pp.1872–1897.
- Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2020) 'Exploring the limits of transfer learning with a unified text-to-text transformer', *Journal of Machine Learning Research*, Vol. 21, No. 140, pp.1–67.
- Rogers, A., Kovaleva, O. and Rumshisky, A. (2021) 'A primer in BERTology: what we know about how BERT works', *Transactions of the Association for Computational Linguistics*, Vol. 8, pp.842–866.
- Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V. and Lanctot, M. (2016) 'Mastering the game of Go with deep neural networks and tree search', *Nature*, Vol. 529, No. 7587, pp.484–489.
- Surbakti, R., Umboh, S.E., Pong, M. and Dara, S. (2024) 'Cognitive load theory: Implications for instructional design in digital classrooms', *International Journal of Educational Narratives*, Vol. 2, No. 6, pp.483–493.
- Sweller, J. (1988) 'Cognitive load during problem solving: effects on learning', *Cognitive Science*, Vol. 12, No. 2, pp.257–285.
- Tucudean, G., Bucos, M., Dragulescu, B. and Căleanu, C.D. (2024) 'Natural language processing with transformers: a review', *Peer J. Computer Science*, Vol. 10, p.e2222.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł. and Polosukhin, I. (2017) 'Attention is all you need', *Advances in Neural Information Processing Systems*, Vol. 30, p.12.
- Wang, Y., He, H. and Tan, X. (2020) 'Truly proximal policy optimization', *Uncertainty in Artificial Intelligence*, Vol. 1, pp.113–122.
- Yngve, V.H. (1960) 'A model and an hypothesis for language structure', *Proceedings of the American Philosophical Society*, Vol. 104, No. 5, pp.444–466.