

Learning Source Representation via Iterative Feature Refinement for Data-Informed Neural Machine Translation

Jinxi Zhang¹, Chunling Li^{2†}

¹ School of Business and Media, Lanzhou University of Finance and Economics, Lanzhou 730020, China.

² School of Statistics and Data Science, Lanzhou University of Finance and Economics, Lanzhou 730020, China.

Abstract

Most existing neural machine translation (NMT) models extract the source representations from lower layers to higher layers without external guidance signals, and the encoder output is used as the final source language representation. In human understanding, the iteration between bottom-up and top-down processes plays a crucial role in comprehension. That is to say, human understanding is full of feedback signals and repeated processing. To obtain a better source language representation, we integrate feedback or recurrent signals in the traditional NMT encoder. This iterative refinement mechanism aligns with the principles of data intelligence, where richer insights are derived through dynamic, context-aware processing of information rather than static, one-pass analysis. To this end, our work presents an approach that is both straightforward and effective. It focuses on learning iterative feature representations from input sentences by means of a recurrent feature encoding (RFE) model. Here, the iterative representation is repeatedly refined based on the feedback context from the previous iteration’s output. Through the proposed RFE, we can progressively obtain a better source representation by the effective interaction between bottom-up and top-down information—enabling the model to exhibit more intelligent behavior in capturing complex linguistic patterns, a key goal of data intelligence. We conduct experiments on the commonly utilized, extensive workshop on statistical machine translation (WMT) 2016 English-to-German (En-De) and WMT 2014 English-to-French (En-Fr) translation tasks. The experimental results show that our proposed iterative feature refinement (IFR) by an RFE significantly outperforms the strong baseline. Moreover, compared with recent deep NMT models and Transformer variants, our approach with Transformer big setting has achieved comparable performance on the two large-scale tasks. Another valuable finding is that we propose a simple Transformer encoder to repeatedly refine the source sentence representations and achieve comparable performance with the state-of-the-art deep NMT models. This demonstrates that enhancing the depth of data interpretation through iterative refinement, rather than merely increasing architectural depth, is a promising direction for advancing data intelligence in natural language processing systems.

Keywords: Neural machine translation; Iterative feature refinement; Recurrent feature encoding; Source representation learning; Data intelligence

1. Introduction

NMT [1][2][3][4][5][6], as a prominent research area in computational linguistics, has garnered substantial attention within the natural language processing (NLP) community [7]. Since the success of the Transformer [8], NMT has achieved impressive performance on many tasks and domains. The recent NMT models [8][9][10] usually consist of a sequence of encoder layers. For instance, the original Transformer model equipped with 6 layers both the encoder and decoder sides progressively extracts feature representations in a feed-forward or bottom-up fashion. As a result, low encoder

[†] Corresponding author: Chunling Li (Email: lichunling@lzu.edu.cn; ORCID: 0009-0007-4271-5214)

layers tend to encode lexical or morphological representations, whereas high encoder layers encode more syntactic and semantic representations [11][12][13]. In the following sections, we use the low-level representations to denote the lexical or morphological representations and high-level representations as the syntactic or semantic representations. Inspired by the idea that robust perception is based on bottom-up and top-down information [14] and its successful application in computer vision tasks [15][16], we conjecture that, in addition to the traditional bottom-up information flow, NMT could also benefit from information flow in a backward or top-down fashion, i.e., the information processing paradigm wherein high-level representations influence low-level representations.

The conveyance of information from both top-down and bottom-up pathways is fundamental to human cognitive functioning and perceptual processes [17][18][19], especially the effective interaction between them. In human cognition, the global context is the key to understanding local inputs with ambiguity or noisy information [20][21]. For instance, in the sentence “*The ‘chip’ on my desk is expensive which I bought in the Apple store.*”, the global semantic representation (from the clause “which I bought in the Apple store”) helps resolve the ambiguity of “*chip*” as computer hardware. From this instance, it was revealed that global context information, through top-down processing, can utilize valuable contextual data to dispel local uncertainty [20]. In addition, with this useful information about the prior useful contexts, when we read the sentence the second or even more times, we will naturally regard the “*chip*” as computer hardware, without ambiguity. This study’s primary objective is to iteratively utilize the wealth of potential top-down or global context information available, in traditional bottom-up or feed-forward NMT encoders. To this end, we carried out experiments with the aim of exploring the cooperative mechanisms of bottom-up and top-down information to alleviate the ambiguity in source sentences, thus achieving better source representation and translation quality.

To allow high-level or global representations to better modulate low-level or local representations, we simply extend the traditional encoder from one-time encoding to a recurrent one. Specifically, we propose a recurrent feature encoding (RFE) model that iteratively learns feature representations from source sentences. In Figure 1, it’s depicted that the RFE model is composed of I homogeneous encoder layers that extract diverse feature representations in a sequential feedforward manner from bottom to top. Meanwhile, the output of the preceding iteration serves as global context information to better guide the current bottom-up process in a top-down fashion (i.e., information flow from the top layer to the bottom layer). Through this iterative feature refinement (IFR) mechanism with an RFE, we process the input iteratively until we get the best source representations for NMT.

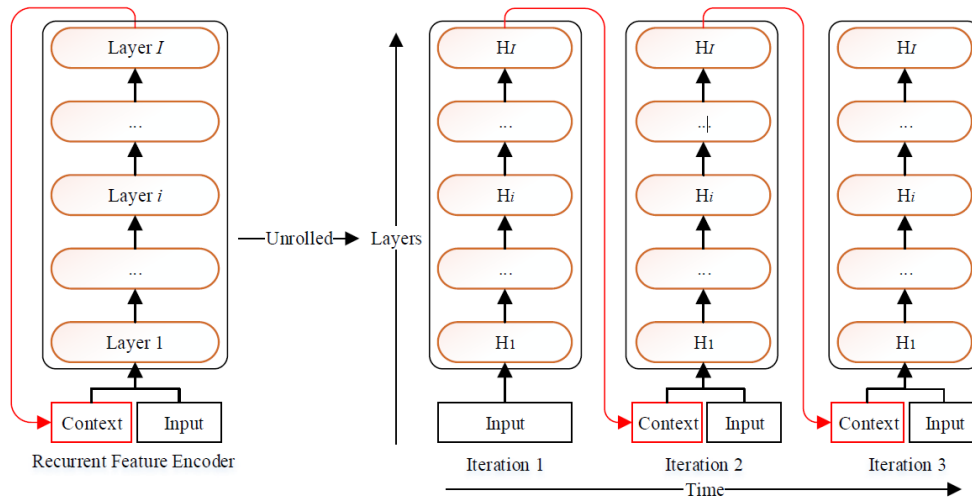


Figure 1: Left: The introduced RFE framework incorporates the preceding iteration’s output into the initial layer as high-level feature input, context information via a top-down feedback connection, and the iteration of RFE can be applied many times. Right: The proposed RFE unrolled in time. We take three iterations as an example to illustrate. The final output of the last iteration is considered as the output of the proposed RFE model.

Empirical investigations were conducted on two major machine translation benchmarks: the WMT 2016 English-German task and the WMT 2014 English-French task, revealing that our proposed IFR mechanism, combined with RFE, significantly enhances translation quality. Additionally, our study offers a comprehensive analysis to interpret the manner in which our IFR mechanism enhances translation performance. To the best of our knowledge, this work is among the first attempts to explore the iterative interaction between high-level and low-level features for encoding source representations in NMT. Overall, this paper presents four major contributions.

- We extend the traditional bottom-up feedforward encoder with a one-time encoding manner into a recurrent manner, which iteratively learns feature representations in a fashion combining both bottom-up and top-down processes.

- We design a simple feedback mechanism in the traditional bottom-up encoder to effectively utilize the wealth of potential global, high-level feature information available. Our approach is simple yet effective.

- Compared with recent state-of-the-art deep NMT models, our RFE-5 model has achieved comparable performance on large-scale WMT English-to-German and English-to-French translation tasks.

- Through the experiments in this paper, we present the potential of a simple yet effective model with appropriate parameter size, which can reach the traditional deep NMT models equipped with huge parameters. This result has a positive effect on alleviating the over-parameter problem in traditional deep neural machine translation models.

This work exemplifies how intelligent data processing—through iterative refinement and context-aware feedback—can elevate the quality of linguistic representations, aligning closely with the core mission of data intelligence: transforming raw input into actionable, contextually enriched knowledge. Our work resonates with recent efforts in iterative refinement [22] and knowledge retention [23], yet uniquely applies these principles within the encoder to enable parameter-efficient, context-aware representation learning.

Although large language models (LLMs) have demonstrated strong multilingual generation capabilities in recent years, our work addresses a different problem context. Specifically, this study focuses on supervised neural machine translation under the standard encoder-decoder framework, with the main goal of improving source-side representation learning within a fixed parameter budget. In this sense, our method should primarily be compared with Transformer-based NMT systems and deep NMT variants, rather than with general-purpose large language models trained on larger datasets and parameter scales. Nonetheless, the iterative refinement concept proposed in this paper may be valuable for future multilingual foundation models and LLM-based translation systems.

2. Related Work

The related work of our proposed approach can be categorized into the following two types.

2.1. Integrating Top-down and Bottom-up Processes

Our central hypothesis is that the integration of top-down and bottom-up signals should be made explicit. As a result, both the high-level and low-level feature representations could modulate each other in the traditional feedforward encoder process. In computer vision, the work of feedback model [24] proposes feedback and skip connections to combine the bottom-up and top-down information. The dual-state recurrent networks [17] exploit both low-resolution and high-resolution signals jointly. In neuroscience, some work [25] explores the integration process of bottom-up and top-down, and the other [19] finds that nonspecific top-down signals can regulate low-order area responses by providing temporal information. In human neuro-imaging, many works have proposed to explore the concurrent use of bottom-up and top-down modulation in the human visual cortex [18][26]. These studies show the effectiveness of bi-directional information flow. Partly inspired by this work, we try to introduce the engagement of high-level features with low-level features in neural machine translation.

Although the question of how to integrate top-down and bottom-up signals has been well studied in computer vision, neuroscience, and neuroimaging, there is little relevant research in NLP. The

work of [20] and [27] propose to mix the pre-trained representations with the bilingual input embedding to fine-tune the NMT model. The universal transformer (UT) model [28] proposes to learn iterative transformations. In the non-autoregressive NMT models[29], the work of [30] and [31] introduces an iterative refinement mechanism for inference. Different from these methods, we aim to obtain a better source representation via an iterative feature refinement mechanism. In addition, [21] proposed to combine both top-down and bottom-up signals in recurrent neural networks (RNNs) for the language model. The work of [20] proposed explicitly incorporating rich latent features to guide NMT encoding. Partly inspired by [20][21], we propose an IFR mechanism for NMT which explicitly integrates both top-down and bottom-up signals. Moreover, we repeatedly refine the fused features of the input, aiming at achieving better performance for NMT.

2.2. Deep NMT Models

Constructing deep NMT models to extract more helpful feature representations serves as a highly efficient method for enhancing translation performance [8][9][32][33][34]. Later, almost all the sophisticated NMT models, including convolutional neural network (CNN)-based models [9][35] and RNN-based models [36][37][38], are based on deep neural architectures. Since the success of the Transformer model [8], many deep NMT models have been proposed, and they have achieved state-of-the-art performance over various tasks [27][39][40][41][42]. All these deep models are equipped with dozens of encoder layers, such as 24 to 60 encoder layers [40][42], or adopt two-stage [27] or multiple-stage [41] training or pre-training to continually grow the encoder depth. An obvious shortcoming of the traditional deep NMT models is that the parameter size increases significantly with the increase in encoder layers. Compared with the most advanced deep NMT model via the shared parameters mechanism [39], our approach is much simpler with little performance loss.

Although we are not trying to build a deep NMT model, the RFE model proposed by us can be regarded as a virtual deep NMT model. The RFE model can keep the parameter size fixed, while flexibly adjusting the model (virtual) depth. The main difference between the RFE and the deep NMT model [39] with shared parameters is that we do not try to fuse intermediate states' features within and among layers' groups. In experiments, our approach achieves comparable performance with the state-of-the-art deep NMT models.

3. Iterative Feature Refinement with Recurrent Feature Encoding

We take the *Transformer* [8] model as our baseline system; here, we only focus on the encoder side. A Transformer encoder is composed of a stack of identical layers. Take the i -th layer as a case in point ($1 \leq i \leq I$), a Transformer encoder layer receives S_i as input state and computes an output state. Note that the input and output sequence are the same length. We use function $ENC(\cdot)$ to denote the encoding process in an encoder layer; please see the original paper [8] for the detailed process. Therefore, as shown in Eq. (1), the information flow in an encoder layer is given by:

$$H_i = ENC(S_i) \quad (1)$$

where H_i denotes the i -th layer output state.

In the Transformer encoder, the i -th layer output will be fed to the $(i + 1)$ -th encoder layer. Specifically, in the first layer, the S_1 is the sum of the source input's word embedding and positional embedding [8]. While in the last layer, H_I is considered as the encoder output, that is, the source sentence representation.

This encoding mechanism shows that the traditional Transformer encoder extracts feature representations in a one-way feed-forward manner, wherein the encoder layer solely obtains feature data from its previous or lower layers. Consequently, the high-level features are extracted based on the low-level features. In this way, the ambiguity of lexical features inevitably appears in the lower encoder layers; moreover, this feature extraction mechanism unavoidably increases the burden of high layers to resolve local uncertainty.

3.1. Recurrent Feature Encoding

To alleviate the load on higher encoder layers, a potential resolution is to deepen the encoder with more layers. This, however, will considerably increase the number of model parameters. Moreover, it requires a carefully designed architecture to overcome the serious gradient vanishing/exploding problems [43][44] caused by stacking more layers [32][33][39].

Partly inspired by the recurrent neural networks [24][45][46] and the latent feature feedback mechanism [20], we propose a recurrent feature encoding (RFE) model to reuse the output of the previous iteration in an iterative manner, as shown in the left part of Figure 1. Our goal is to reuse the previous iteration’s output as an implicit global context representation and integrate it into the source input, resulting in the global contexts enhanced source input. Then, we feed the mixed input to the current RFE iteration. In this way, we integrate high-level feature information into the low-level input features of the traditional bottom-up encoder, so that they can interact with each other. As a result, even the lowest layer receives both global semantic representations and local word embeddings, making it capable of resolving ambiguity in low encoder layers. We will give a detailed linguistic analysis of this problem in Section 5.1.

To this end, we use H_i^t to denote the i -th layer output in t -th iteration ($1 \leq t \leq T$) of the RFE. Consequently, for i -th layer in t -th iteration, the transformation in Eq. (2) is updated as:

$$H_i^t = ENC(S_i^t) \quad (2)$$

For a special case, in the first layer, S_1^t is consistently assigned to the source input.

We modify the one-time directional encoder process into a recurrent manner; thus, the current encoder output can be used as a useful global context to guide the next encoding iteration. We do not change the architecture of the Transformer encoder layer and thus do not introduce any additional parameters. This is quite different from related studies that propose sophisticated recurrent structures to construct recurrent neural networks, such as recurrent CNNs[45], UT [28] and Top-Down Networks [15]. Moreover, compared with the work of [20], our proposed approach does not need to pre-train the feature encoder. Thus, the proposed approach is simple and flexible.

3.2. Iterative Feature Refinement

To extract a better representation of the source through the RFE model, we propose an iterative feature refinement mechanism to dynamically and iteratively encode the input of the source. For the t -th iteration ($t > 1$) of an RFE, we use H_i^{t-1} , i.e., the output of the previous iteration as global representations to guide the input of the current iteration, as shown in the right part of Figure 1. Though there are I encoder layers in each iteration, we simply feed hidden states H_i^{t-1} to the first layer of the current iteration, while keeping the other $I - 1$ layers receiving information from their corresponding previous layer only.¹ Accordingly, the information transformation of the first layer in Eq. (3) is updated as:

$$H_1^t = ENC(f(S_1^t, H_1^{t-1})) \quad (3)$$

$$f(S_1^t, H_1^{t-1}) = S_1^t + H_1^{t-1} \quad (4)$$

where $f(\cdot)$ is a fusion function, which fuses the source input and the feedback context. Note that S_1^t is consistently assigned to the source input embedding. The Eq. (4) realizes the interaction between the low-level information S_1^t and high-level context H_1^{t-1} , which is one of the key points of our approach. In this way, the fused representations of high-level features and source embedding are re-fed to the encoder to encode the source feature representations. We will give further discussion on the interaction between them in Section 5.1 and 5.2. For Eq. (4), we also considered alternative fusion functions, including mean pooling and a gating mechanism [47]. Their performance is comparable to that of simple addition in our preliminary experiments; for completeness, the quantitative comparison is now provided in Appendix A (Table A1). Since simple addition yields competitive accuracy while introducing no extra fusion parameters and minimal computational

¹ While it is also practicable to directly feed H_i^{t-1} back into all layers in the current iteration, our preliminary experimental result shows this achieves similar translation performance.

overhead, we adopt it as the default choice in this paper.

Finally, we use H_t^f , i.e., the output from the top encoder layer of the final RFE iteration, as the ultimate output of the RFE. It should be noted that during the initial iteration ($t = 1$), the RFE functions in the manner of a conventional Transformer encoder. Through this iterative encoder process, we have realized a simple yet effective integration of high-level and low-level information in a single encoder. By applying the IFR multiple times, we can progressively obtain better source language representations. This iterative, self-improving mechanism embodies a fundamental principle of data intelligence: the capacity to derive deeper meaning through repeated, context-sensitive analysis of data, rather than relying on static, single-pass transformations.

Unrolling the RFE through time will produce a deep neural network with a fixed parameter size. Thus, the RFE proposed by us can be regarded as a virtual deep NMT model. The grouping skipping connection model [39] proposed a deep NMT model with shared parameters, and the main differences between our RFE and the deep NMT model are as follows: 1) We repeatedly refine the fused features of the input, instead of trying to fuse the different features of intermediate layers. 2) We fuse the output of the previous RFE iteration with the source input, and then re-feed it into the encoder, while the deep NMT with shared parameters repeatedly feeds all the previous features into each layer of the current group. 3) We take the final output of the last iteration as the output of the proposed RFE, while the deep NMT model fuses different layers to get the encoder output. Therefore, our proposed RFE is a single encoder model, which encodes the input source language representations several times and gets the final source representations at the last iteration.

3.3. Training and Inference

In the training stage, both the proposed RFE model and the Transformer decoder are jointly trained on the training data. Note that we do not change the architecture of the Transformer decoder and the training objective function. Therefore, the proposed RFE model maintains a simple end-to-end architecture. During the inference phase, we also apply RFE multiple times to get better source representations. By default, the number of iterations in an RFE model is the same during training and inference.

4. Experiments

Within this section, we conduct comprehensive experiments on the WMT 2016 En-De and WMT 2014 En-Fr tasks. To illustrate the advantages of our model, all our experiments are carried out on the mainstream, large-scale machine translation datasets, and the proposed RFE model is implemented in the open-source translation system.

4.1. Data Sets and Evaluation

The details of datasets and evaluation metrics in our experiments are listed as follows: En-De translation: We carry out experiments on the WMT16 En-De data set, which is also used in the previous related work, so we can make a fair comparison with the previous studies. The 2016 WMT English-German training dataset is consistent with the earlier related work [8][27][40][41][48][49]. We utilize the processed data supplied by the original Transformer [8]². In this data set, sentences are tokenized and processed utilizing a collaborative byte-pair encoding (BPE) method involving 32K procedures. The training data has 4.5M sentences for training and 3000 sentences for both validation and evaluation. We use newstest2013, newstest2014 as the development set and the test set, respectively.

En-Fr translation: We choose the standard benchmark corpus from the WMT 2014³ as the En-Fr training data. We take newstest2013, newstest2014 as the development set and the test set, respectively.

Following previous work [50], all the sentences undergo tokenization using the canonical Moses scripting protocols [51]. Then, we apply a joint BPE⁴ with 32K operations to segment words into

² https://drive.google.com/uc?export=download&id=0B_bZck-ksdkpM25jRUN2X2UxMm8

³ <http://www.statmt.org/wmt14/translation-task.html>

⁴ <https://github.com/rsennrich/subword-nmt>

sub-word units. Finally, we remove the long sentences by setting a maximum sentence length of 256 tokens. After pre-processing steps, the En-Fr training corpus contains 35.6M training sentences. The pre-processing scripts for the En-Fr dataset used in this paper are provided by *Fairseq* ⁵.

To test the performance of our experiments, we choose the strictly case-sensitive BLEU score measured by SacreBLEU score (SacreBLEU) [52] to achieve equitable comparison with relevant studies. Simultaneously, in accordance with prior research, we likewise present BLEU scores ascertained through the multi-bleu assessment.perl scripts on both En-De and En-Fr translation tasks. Our evaluation metrics can be compared fairly and clearly with the existing or related work. We report SacreBLEU (SBLEU) scores.

4.2. Model Settings

Our approach adopts the *Fairseq* [50] as the foundational Transformer system ⁶ since most of the related systems are based on this system. We choose the default model settings for both Transformer base and big models [50] in all our experiments. For model training, the Adam optimizer (β_1 is 0.9, β_2 is 0.997, learning rate = 0.001) is used to train all the NMT models. We adopt the learning rate decay strategy put forward by [8] to stabilize training. In the configuration detailed, both the base and big models are assigned a warm-up phase of 4000 iterations, while the label smoothing parameter is designated as 0.1. Our models are trained on four Tesla V100 GPUs. We set the gradient update frequency as 2, and the token size is 4096. To successfully train the big models, we will reduce batch size to fit the GPU memory, while increasing the update frequency accordingly. All our systems are trained for 100K/300K update steps for base/big models on the En-De translation task. For the En-Fr task, we trained 300K/300K for base/big models.

For inference, we average the last 5/10 epoch checkpoints for base/big models to get the final single model on the En-De and En-Fr experiments. This model averaging method is also used in related work. We set the beam size to 6 and the length penalty to 0.6 for all the En-De and En-Fr experiments. For En-De experiments, we report the BLEU scores by averaging the 3 experiments, which are run three times with three distinct random seeds. Regarding the En-Fr translation assignment, all the systems are only run once. Since the En-Fr corpus is large, it needs a huge amount of computational cost.

4.3. Main Experimental Results

In this paper, the terms “Transformer” and “Transformer-12” denote the models that have 6 or 12 encoder layers, respectively. In the next sections, the Transformer-12 baseline can be considered as a special case of RFE, i.e., an RFE with 1 iteration. “RFE- X ” denotes applying X iterations for an RFE, while all RFEs have 12 encoder layers. In our preceding experiments, we have observed that when $I = 12$, the RFE model gets the best performance while the model parameters’ size is in a suitable range. This is also an interesting finding; in some previous work [39][40], it is found that grouping 12 Transformer encoder layers together is a good practice in NMT. Therefore, all RFEs and baselines in our experiments adopt a 12-layer encoder. Note that all our experiments have 6 decoder layers, by default.

We take our main experiments on Transformer big setting and evaluate it in relation to the most advanced deep NMT systems and Transformer variant models. Following previous work [39][40][41][50], we run our best system (RFE-5) in the context of En-De and En-Fr translation tasks directions on 4 GPUs with an update frequency of 16 and set a larger learning rate of 0.005 for better convergence. In order to ensure a fair comparison, we revise the RFE-5 systems of 30K and 50K correspondingly for En-De/En-Fr tasks. The main experimental results are as follows:

1) Performance on En-De and En-Fr Translations. As shown in Table 1, it is observed that our re-implemented Transformer systems are strong on both En-De and En-Fr tasks. The performance of the Transformer-12 baseline is significantly higher than that of the Transformer. Since our NMT systems adopt pre-normalization [33] in each sub-layer, doubling the number of encoder layers is helpful for translation performance. This structure is widely applied in recent deep NMT models [39][40][41].

⁵ <https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-wmt14en2fr.sh>

⁶ <https://github.com/pytorch/fairseq>

Our proposed RFE-5 systems gain 1.4 and 1.1 SBLEU on En-De and En-Fr tasks over the baseline systems respectively, indicating that our strategy substantially outperforms the strong baseline. This demonstrates that our RFE model significantly enhances translation performance.

2) Comparison with the most advanced Models. In accordance with [24][39], the encoder layers’ depth is regarded as physical depth, while the number of iterations linked to physical depth is treated as virtual depth. Thus, unfolding the RFE through time will produce a deep neural network with a fixed parameter size. Thus, we compare our proposed RFE model to various state-of-the-art deep NMT models.

As shown in Table 1, it is observed that our proposed RFE-5 system (virtual depth of 60) has reached the forefront results in En-Fr translation tasks. Besides, the performance of our proposed RFE-5 is close to the state-of-the-art deep NMT models [39] on the En-De task. This finding is highly encouraging as the parameter size of our model remains unchanged and the performance is achieved by a single encoder. Furthermore, expanding the encoder layer count in conventional deep architectures inevitably introduces substantial over-parameterization problems [40][41][42], or requires two-steps [27] or multiple-steps [41] training phases with thoughtful reinstating learning rates and commencing parameter initialization [42]. As an alternative, we employ the conventional end-to-end training approach within a single phase.

Table 1: The BLEU scores for the proposed method employing the Transformer big model configuration in the En-De and En-Fr translation tasks. Hereafter, †/‡: significant over the baseline Transformer-12 at 0.05/ 0.01. “#Para.” represents the parameter count (million) while “Speed” refers to the training speed (second/updates). “BLEU” and “SBLEU” are for scores evaluated by multi-bleu.perl and SacreBLEU, respectively.

<i>Systems</i>	<i>WMT16 En-De</i>				<i>WMT14 En-Fr</i>			
	Para.	Speed	BLEU	SBLEU	Para.	Speed	BLEU	SBLEU
<i>Existing Systems</i>								
Transformer[8] (big)	213.0M	n/a	28.4	n/a	222.1M	n/a	41.3	n/a
Scaling NMT[50] (big)	210.0M	n/a	29.3	28.6	222.0M	n/a	43.2	41.4
Evolved Transformer[48] (big)	218.0M	n/a	29.8	29.2	n/a	n/a	n/a	n/a
Layer Drop[53] (big)	286.0M	n/a	30.2	n/a	n/a	n/a	n/a	n/a
Deep Transformer[49] (deep)	287.0M	n/a	n/a	28.7	n/a	n/a	n/a	n/a
SDT-PRR-48L[41] (deep)	194.0M	n/a	30.2	29.0	199.0M	n/a	43.3	41.5
Deep NMT[42] (deep)	256.0M	n/a	n/a	29.5	262.0M	n/a	43.8	41.8
Depth Growing[27] (big)	270.0M	n/a	29.9	n/a	281.0M	n/a	43.3	n/a
MSC[40] (big)	512.0M	n/a	30.6	n/a	n/a	n/a	n/a	n/a
SDT-PRR-24L[41] (big)	437.0M	n/a	30.5	29.3	n/a	n/a	n/a	n/a
Multi-Groups-60[39] (big)	296.2M	n/a	30.7	29.6	n/a	n/a	n/a	n/a
<i>Our Systems</i>								
Transformer (big)	214.3M	1.2	28.7	27.6	215.1M	1.2	41.9	40.1
Transformer-12 (big)	289.8M	1.7	29.2	28.1	290.6M	1.7	42.6	40.8
RFE-5 (big)	289.8M	3.6	30.6	29.5†	290.6M	3.6	43.8	41.9‡

Compared with the advanced Transformer variant models [48][50][53] or deep Transformer models [49] with 96 layers, our RFE-5 system achieves better performance with an ordinary Transformer encoder equipped with a simple recurrent structure. Meanwhile, compared with the depth growing approach [27], our RFE-5 achieves higher performance with slightly more parameters. Contrasted with the deep NMT model involving parameter sharing [39], our proposed RFE model does not need to carefully integrate the previous features, so it is a much simpler approach to implement.

From the experimental results, we can see that compared with the most advanced deep NMT models, our proposed RFE model exhibits competitiveness with respect to both parameter magnitude and translation performance. This demonstrates that data intelligence in NLP need not stem solely from architectural complexity, but can emerge from intelligent data usage—refining representations iteratively to extract maximal insight from fixed model capacity.

4.4. Performance with Transformer Base Model Setting

Table 2 illustrates the performance of our proposed RFE within the configuration of the Transformer base model. It shows that introduced RFE model demonstrates a notable enhancement compared to the baseline system across the tasks of En-De and En-Fr.

Table 2: BLEU scores attained by our recommended method under the Transformer base model configuration for En-De and En-Fr translation tasks.

Systems	Para.	Speed	BLEU	SBLEU
<i>WMT16 En-De</i>				
Transformer	63.0M	0.6	27.3	26.3
Transformer-12	82.0M	0.8	28.3	27.2
RFE-3	82.0M	1.1	28.8	27.7†
RFE-5	82.0M	1.5	29.4	28.3‡
<i>WMT14 En-Fr</i>				
Transformer	63.3M	0.6	39.9	38.1
Transformer-12	82.3M	0.8	40.9	39.1
RFE-3	82.3M	1.1	41.6	39.8†
RFE-5	82.3M	1.5	42.0	40.2‡

In Table 2, it can also be observed that:

- When repeating the RFE 3 iterations (i.e., RFE-3), we observe significant improvement over Transformer-12 within the contexts of the En-De and En-Fr translation. Meanwhile, if we continuously extend the RFE from 3 to 5 iterations, we still observe further significant improvement. This indicates that even with no additional parameters, repeating the proposed RFE is helpful to improve translation quality. We will give a fine-grained analysis on the iterations of RFE on En-De translation in Section 5.3.
- Our proposed RFE-5 system achieves the best performance of 28.3 (40.2) SBLEU scores on the test set of En-De (En-Fr), with 1.1 (1.1) SBLEU scores significant enhancement compared to the robust Transformer-12 baseline. This shows that the IFR mechanism with multiple iterations can continuously and significantly improve translation performance.
- Characterized by a lower parameter quantity and analogous computational duration, our proposed RFE-5 even outperforms the Transformer-big model (as shown in Table 1), e.g., 28.3 in SBLEU vs. 27.6 for En-De and 40.2 vs. 40.1 for En-Fr.
- The efficacy of the RFE model in the En-De task aligns with that in the En-Fr task. This indicates that the performance of our proposed RFE model is very robust over two large-scale data sets.

As is evident from the aforementioned, our method has achieved consistent performance on both the big and base models, which shows that this approach has good generalization.

To further compare with mainstream autoregressive architectures, we also include a lightweight comparison with two representative non-Transformer baselines, specifically the attention-based recurrent NMT model (RNNSearch+Attn) and the convolutional sequence-to-sequence model (ConvS2S). As shown in Table 3, models based on RNNs and CNNs performed worse than Transformer and RFE models in the WMT16 En-De experiment. This observation further supports the effectiveness of the proposed iterative refinement mechanism under modern NMT architectures and demonstrates superior performance compared to representative autoregressive baselines.

Table 3. Supplementary comparison of representative autoregressive RNN/CNN-based NMT models on WMT16 En-De. The results of the RNN/CNN baselines are reproduced or adapted from the settings commonly reported in previous work under similar preprocessing and evaluation protocols.

Model	Architecture	BLEU	SBLEU
RNNSearch+Attn	RNN-based	26.4	25.3
ConvS2S	CNN-based	27.1	26.0

4.5. Parameters and Speed

As illustrated in Tables 1 and 2, in contrast to the Transformer-12 baseline system, the proposed RFE model does not introduce additional parameters. Although no parameters are added in our models, applying the RFE multiple times will bring more computation costs. Nevertheless, when juxtaposed with the Transformer big model, our proposed RFE-5 on the base model achieves superior performance while utilizing fewer parameters and maintaining a comparable computational cost. Due to the effective computation of our fusing mechanism in Eq. (4), with the increase of iterations, the computation cost of our proposed RFE is increased slightly. Overall, the updated cost of the RFE-5 model is about 2 times that of the baseline model.

During the model inference phase, the proposed RFE model leads to a slight increase in inference time. For instance, in the base model configuration, RFE-5 results in a 9.3% increase in decoding time, while in the big model configuration, it causes a 29.7% increase, relative to the baseline system. The duration of decoding is calculated through the mean of three decoding attempts on the En-De test set.

These results suggest that the proposed IFR mechanism improves translation quality mainly by increasing model capacity with a fixed parameter budget, rather than by reducing computational overhead. In other words, the key advantage of RFE is parameter efficiency, not free performance gains. RFE-5 achieves higher BLEU/SBLEU scores without increasing the number of parameters compared to Transformer-12, but this advantage comes at the cost of additional iterative computations during training and inference. Therefore, the proposed method should be understood as a parameter-efficient optimization strategy with measurable latency trade-offs.

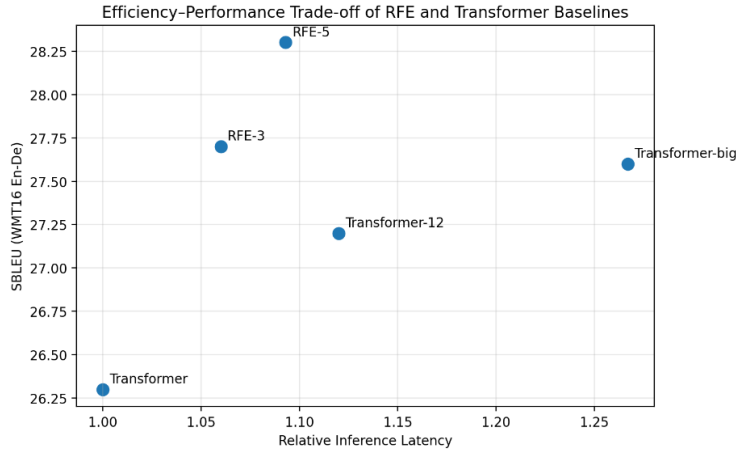


Figure 2: Efficiency-performance trade-offs for RFEs and Transformer baselines on the WMT16 English-German dataset. The plot plots the relative inference latency versus the SBLEU of the WMT16 English-German dataset. The Transformer is used as a reference point. The proposed RFE-based model offers improved translation quality with modest additional latency, illustrating the trade-off between performance gains and iterative computational overhead.

To further visualize this trade-off, Figure 2 plots the relative inference latency versus SBLEU on the WMT16 English-German data. It can be observed that RFE-3 and RFE-5 have significantly improved translation quality compared with Transformer and Transformer-12, while only increasing moderate decoding delay. In particular, RFE-5 is in the favorable area of the efficiency-performance space: although it is slightly slower than the baseline model, it provides the highest translation quality

among the contrasted parameter-matched models. This suggests that this approach is best suited for scenarios where translation quality is prioritized under a fixed parameter budget.

5. Analysis and Discussion

In this part, we conduct linguistic probing tasks and ablation experiments on the English-to-German (En-De) translation tasks and further discuss how the iterative feature refinement mechanism we propose can improve the translation quality.

5.1. Linguistic Analysis

The function of the NMT encoder is extracting useful feature representations from input sentences. Through linguistic probing tasks, we can see how many of these linguistic features could be encoded in the hidden states [20][39][54]. Here, we carry out probing experiments to acquire linguistic insights regarding the diverse types of linguistic features encoded within the hidden states of the NMT encoder. Then, we explain how NMT could benefit from our proposed IFR mechanism through the linguistic features encoded in the encoder layers. We follow the work of [54] to probe the linguistic features embedded in the first layer output in different iterations of the RFE-5 model, as shown in Figure 3. Following the previous work [39][55][56], we take a mean pooling operation on the hidden states of an encoder layer output to obtain the source sentence representation.

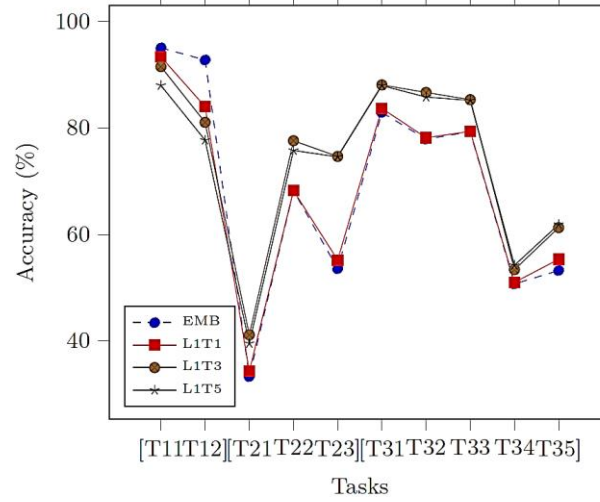


Figure 3: Precision in the probing tasks conducted on the encoder of the RFE-5 model. “EMB” represents the embedding layer, and “L1T X ” signifies the initial layer output within the X -th iteration. Hereafter, we use T11, T12 to denote the two surface tasks, T21, T22, T23 to denote the three syntactic tasks, and T31, T32, T33, T34, T35 to denote the five semantic tasks.

The ten linguistic probing tasks can be divided into three types: surface features, syntactic features, and semantic features probing tasks. The surface features task includes the sentence length (SeLe) and word content (WC) tasks. The syntactic features task contains the bigram shift (BShif), tree depth (TrDep), and top constituent (ToCo) tasks. The evaluation of semantic features integrates the tasks related to tense (Tense), subject number (SubN), object number (ObjN), semantic anomaly detection (SoMo), and coordination reversal (CoIn). For each of these probing tasks, we adhere to the default configurations, utilizing the provided datasets, and the data sets are provided by the original work.⁷

From Figure 3, we find that: 1) the embedding layer contains the least syntactic and semantic information, but the most surface features. Since the embedding layer state has not been processed by the NMT encoder. 2) the linguistic features (except the T12 task) contained in the first layer output in the first iteration are almost consistent with the embedding layer. Since the first layer has undergone only one process, thus the encoded features are not sufficient. 3) After applying multiple

⁷ <https://github.com/facebookresearch/SentEval>

RFE iterations, the first layer output encompasses a greater number of syntactic and semantic characteristics, whereas it is noted that the surface-level features diminish rapidly. This is in line with our conjecture that high-level contextual information (feedback from the previous RFE iteration) can reduce ambiguous surface features or enhance the syntactic and semantic features in low layers. These findings show that the IFR mechanism we propose is effective, which is also in line with the human understanding process.

5.2. Ablation Study

As shown in Figure 1 above, except for the first RFE iteration, each iteration receives two types of input information, i.e., the feedback context representations from the previous RFE iteration and the source sentence input. Here, we reuse the source input embedding to facilitate the interplay among high-level contextual representations and low-level source input representations. In this way, we could build iteratively refined feature representations through more encoding iterations. To test the effect of reusing the source input, we compare RFE-5 with and without source input reuse.

As shown in Table 4, by removing the source input from iterations of 2 to 5, the translation quality decreases significantly from 28.3 to 27.4, and is close to the performance of the Transformer-12 baseline system. This shows that the IFR mechanism of integrating high-level and low-level feature representation is the key point of our proposed RFE model.

Table 4: Translation performance on the test set of En-De translation when reusing or not reusing the source input.

Systems	Para.	SBLEU
Transformer-12	82.0M	27.2
RFE-5	82.0M	28.3
-Input	82.0M	27.4

5.3. Analysis on Recurrent Feature Encoding

A significant hyper-parameter (T in Eq. (2)) in our RFE model is the number of iterations. We compare six models with iteration counts ranging from 1 to 6 to test the influence of different iterations on translation quality.

As shown in Figure 4, the translation quality is improved significantly when increasing the iteration number from 1 to 5, suggesting that the NMT encoder benefits from more RFE iterations in capturing refinement features from the source input. The model with 5 iterations achieves the highest SBLEU scores, while further scaling up to 6 even slightly hurts translation performance. It shows that there is still room for improvement in how to continuously improve translation quality through IFR. Moreover, it also indicates that how to continuously extract helpful feature representations from the source input is a problem worthy of further study.

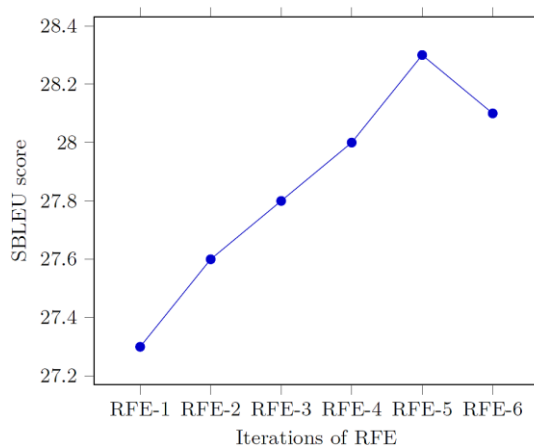


Figure 4: Translation quality with respect to the iterations of RFE. RFE-1 denotes the Transformer-12 baseline system.

5.4. Comparison with Deep Transformer Model

Figure 5 presents translation quality with respect to different encoder depths. The performance of both the RFE and the original Transformer model benefits from increasing encoder depth from 12 to 24. This shows that a greater increased model depth encoding process could improve the translation quality.

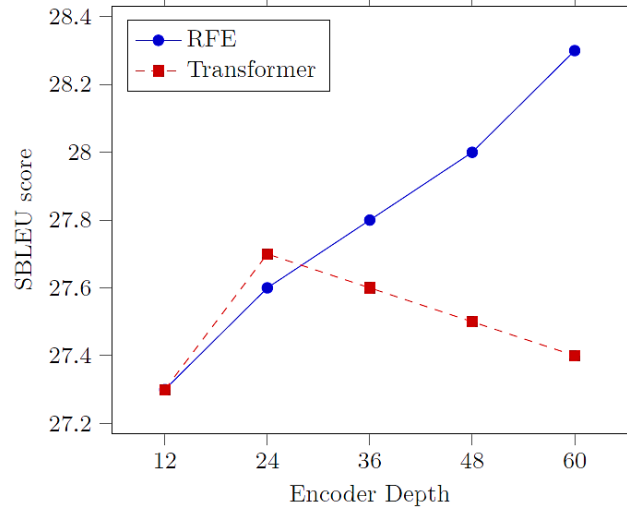


Figure 5: Translation quality with respect to different encoder depths. Transformer denotes the original Transformer baseline system.

For the Transformer baseline, when the encoder layers exceed 24, it will hurt the performance, although we can successfully train a 60-layers Transformer model. This shows that simply increasing the number of encoder layers does not lead to continuous extraction of useful features for improving the translation performance. This result confirms the finding of previous work [33][39][57].

Unlike the Transformer baseline models, the RFE model proposed by us can continuously improve the translation quality from 12 to 60 (virtual) encoder depth. It can be seen that our approach outperforms the traditional deep Transformer model while keeping the parameter size unchanged. In addition, it also shows that our RFE model can obtain more useful features to improve the translation quality.

5.5. Results on Various Lengths of Sentence

Short sentences usually lack context, while long sentences tend to contain more context information. Thus, in the NMT encoder, long sentences can provide more useful context, and short sentences usually need more context, to eliminate ambiguity. We conjecture that both the short and long sentences benefit from the RFE mechanism because it encompass broader global contextual information repeatedly. Following the previous work [58][59][60], we categorize the source sentences within the test set based on their lengths and compute their BLEU scores individually.

Figure 6 shows the BLEU scores on different sentence lengths. From the experimental results, we find that our RFE model is superior to the Transformer-12 baseline system over all sentence lengths.

We additionally observe that the performance on different sentence lengths is different. For instance, the discrepancy in performance between the Transformer-12 model and the RFE approach we propose is more pronounced for both brief sentences (comprising fewer than 30 words) and extended sentences (containing over 50 words).

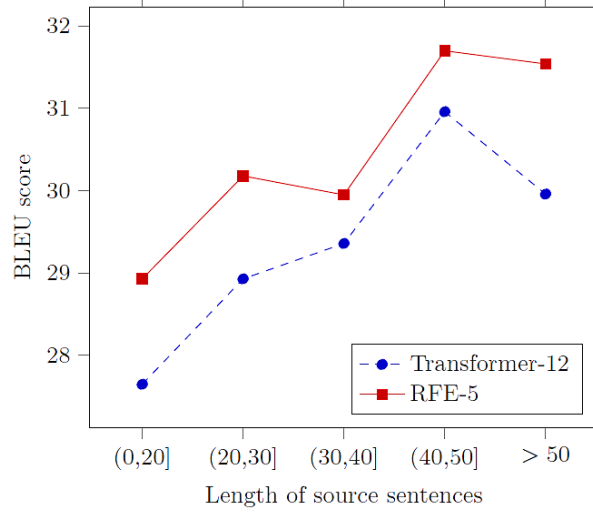


Figure 6: BLEU scores on the different lengths of sentences in the test sets. Note that we compute the scores by the multi-bleu.perl script.

6. Conclusion

In the pursuit of data intelligence, where systems are expected to extract meaningful, context-aware insights from complex linguistic data, effective representation learning is paramount. In order to obtain a better representation of the source language through a single encoder, we propose an iterative feature refinement model for NMT. Specifically, we propose an RFE model to extract useful feature representations from the source sentence and feed them back to the next RFE iteration through the top-down feedback connections. Aiming at continuously improving the quality of source representations, we proposed to apply the RFE multiple times to gradually learn better source representations. Based on our current understanding and available knowledge, this work presents an initial exploration of the interaction between top-down and bottom-up processes for encoding source representations in NMT. We conduct experiments on the widely used, large-scale WMT 2016 English-to-German and WMT 2014 English-to-French translation tasks. The empirical findings demonstrate that our proposed RFE model attained a substantial enhancement compared to the strong Transformer system. Compared with the advanced traditional deep NMT systems, our approach achieves comparable performance on the En-De and En-Fr translation tasks. Notably, our approach is simple and very easy to implement.

Although the input information is static, human understanding is a dynamic process with recurrence and feedback. Therefore, understanding is a process of continuous processing and refinement of the static input information. In forthcoming research endeavors, we plan to investigate more efficacious IFR for various NMT models [61]. In addition, it is worth exploring whether the proposed iterative optimization strategy can be applied to multilingual foundation models or LLM-based translation systems, in which cases encoder-side context optimization may still offer benefits under parameter-efficient adaptation settings. Meanwhile, we will evaluate our RFE systems against other NLP tasks, such as abstractive text summarization [62] and sentiment analysis. In addition, we will explore the use of a single physical encoder by repeatedly refining the source input features to outperform the traditional deep NMT models equipped with a huge parameter size. Future work may explore hybrid approaches combining our IFR mechanism with knowledge distillation frameworks such as CoRe-KD [23], potentially enabling even more efficient transfer of linguistic knowledge in low-resource or continual learning settings.

Author Contributions

Jinxi Zhang: contributed the conceptualization, methodology, formal analysis and writing.

Chunling Li: conceptualization; supervision; supervised the project and revised the manuscript.

Acknowledgements

Thank you to my colleagues and graduate students for their guidance and assistance, and to the editors and reviewers for their valuable feedback.

Data Availability Statement

All datasets are publicly available. Our implementation of the proposed RFE/IFR model is based on Fairseq and is publicly available at: <https://github.com/happy0happy606/NMT/>

References

- [1] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014. [Online]. Available: <https://peerj.com/articles/cs-2607/code.zip>.
- [2] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, 2014, pp. 103–111. [Online]. Available: <https://aclanthology.org/W14-4012.pdf>.
- [3] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proceedings of Neural Information Processing Systems*, 2014, pp. 3104–3112. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2014/hash/5a18e133cbf9f257297f410bb7eca942-Abstract.html.
- [4] L. Kang, S. He, M. Wang, F. Long, and J. Su, "Bilingual attention based neural machine translation," *Applied Intelligence*, vol. 53, no. 4, pp. 4302–4315, 2023. [Online]. Available: <https://link.springer.com/article/10.1007/s10489-022-03563-8>.
- [5] Y. Yin, J. Zeng, J. Su, C. Zhou, F. Meng, J. Zhou, D. Huang, and J. Luo, "Multi-modal graph contrastive encoding for neural machine translation," *Artificial Intelligence*, vol. 323, p. 103986, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0004370223001327>.
- [6] Y. Li, J. Li, J. Jiang, S. Tao, H. Yang, and M. Zhang, "P-transformer: Towards better document-to-document neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 3859–3870, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10255243/>.
- [7] NLLB Team, "Scaling neural machine translation to 200 languages," *Nature*, vol. 630, pp. 841–846, 2024. [Online]. Available: <https://www.nature.com/articles/s41586-024-07335-x>.
- [8] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of NIPS*, 2017, pp. 5998–6008. [Online]. Available: <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html>.
- [9] J. Gehring, M. Auli, D. Grangier, and Y. N. Dauphin, "A convolutional encoder model for neural machine translation," in *Proceedings of ACL 2017*, 2017, pp. 123–135. [Online]. Available: <https://aclanthology.org/P17-1012/>.
- [10] J. Zhou, Y. Cao, X. Wang, P. Li, and W. Xu, "Deep recurrent models with fast-forward connections for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 4, pp. 371–383, 2016. [Online]. Available: <https://aclanthology.org/Q16-1027/>.
- [11] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 2227–2237. [Online]. Available: <https://aclanthology.org/N18-1202/>.
- [12] A. Poliak, Y. Belinkov, J. Glass, and B. Van Durme, "On the evaluation of semantic phenomena in neural machine translation using natural language inference," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2018, pp. 513–523. [Online]. Available: <https://aclanthology.org/N18-2082/>.
- [13] W. Ma, S. Liu, M. Zhao, X. Xie, W. Wang, Q. Hu, J. Zhang, and Y. Liu, "Unveiling code pre-trained models: Investigating syntax and semantics capacities," *ACM Transactions on Software Engineering and Methodology*, vol. 33, no. 7, pp. 1–29, 2024. [Online]. Available: <https://dl.acm.org/doi/full/10.1145/3664606>.
- [14] C. Frenkel, D. Bol, and G. Indiveri, "Bottom-up and top-down approaches for the design of neuromorphic processing systems: Tradeoffs and synergies between natural and artificial intelligence," in *Proceedings of the IEEE*, vol. 111, no. 6, pp. 623–652, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10144567>.
- [15] I. Lelekas, N. Tomen, S. L. Pintea, and J. C. van Gemert, "Top-down networks: A coarse-to-fine reimagination of CNNs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 3244–3253. [Online]. Available: https://openaccess.thecvf.com/content_CVPRW_2020/html/w45/Lelekas_Top-Down_Networks_A_Coarse-to-Fine_Reimagination_of_CNNs_CVPRW_2020_paper.html.
- [16] V. Ramanishka, A. Das, J. Zhang, and K. Saenko, "Top-down visual saliency guided by captions," in *Proceedings of CVPR*, 2017, pp. 7206–7215. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Ramanishka_Top-Down_Visual_Saliency_CVPR_2017_paper.html.
- [17] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang, "Image super-resolution via dual-state recurrent networks," in *Proceedings of CVPR*, 2018, pp. 1654–1663. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Han_Image_Super-Resolution_via_CVPR_2018_paper.html.
- [18] S. J. D. Lawrence, D. G. Norris, and F. P. de Lange, "Dissociable laminar profiles of concurrent bottom-up and top-down modulation in the human visual cortex," *Neuroscience*, vol. 8, pp. 1–10, 2019. [Online]. Available: <https://elifesciences.org/articles/44422>.
- [19] J. H. Lee, M.-H. Kim, and S. Vijayan, "Temporal learning of bottom-up connections via spatially nonspecific top-down inputs," *Neurocomputing*, vol. 411, pp. 128–138, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0926641020300000>.

- <https://www.sciencedirect.com/science/article/abs/pii/S0925231220310018>.
- [20] Y. Li, J. Li, and M. Zhang, "Improving neural machine translation with latent features feedback," *Neurocomputing*, vol. 463, pp. 368–378, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/abs/pii/S0925231221012030>.
 - [21] S. Mittal, A. Lamb, A. Goyal, V. Voleti, M. Shanahan, G. Lajoie, M. Mozer, and Y. Bengio, "Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules," in *Proceedings of the 37th International Conference on Machine Learning, PMLR*, 2020, pp. 6972–6986. [Online]. Available: <https://proceedings.mlr.press/v119/mittal20a.html>.
 - [22] S. Chen, X. Shi, P. Li, J. Liu, and Y. Li, "Refining translations with large language models: A constraint-aware iterative prompting approach," *Data Intelligence*, pp. 1–33, 2025. [Online]. Available: <https://www.sciengine.com/DI/doi/10.3724/2096-7004.di.2025.0130>.
 - [23] Z. Lin and D. Huang, "CoRe-KD: Contrastive multi-teacher and retention-aware knowledge distillation for continual multilingual neural machine translation," *Data Intelligence*, pp. 1–15, 2025. [Online]. Available: <https://www.sciengine.com/DI/doi/10.3724/2096-7004.di.2025.0109>.
 - [24] A. R. Zamir, T. Wu, L. Sun, W. Shen, J. Malik, and S. Savarese, "Feedback networks," in *Proceedings of CVPR*, 2017, pp. 1308–1317. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2017/html/Zamir_Feedback_Networks_CVPR_2017_paper.html.
 - [25] A. M. Bastos, J. Vezoli, C. A. Bosman, J.-M. Schoffelen, R. Oostenveld, J. R. Dowdall, P. De Weerd, H. Kennedy, and P. Fries, "Visual areas exert feedforward and feedback influences through distinct frequency channels," *Neuron*, vol. 85, no. 2, pp. 390–401, 2015. [Online]. Available: [https://www.cell.com/neuron/fulltext/S0896-6273\(14\)01099-X](https://www.cell.com/neuron/fulltext/S0896-6273(14)01099-X).
 - [26] N. Dijkstra, P. Zeidman, S. Ondobaka, M. A. J. van Gerven, and K. Friston, "Distinct top-down and bottom-up brain connectivity during visual perception and imagery," *Scientific Reports*, vol. 7, p. 5677, 2017. [Online]. Available: <https://www.nature.com/articles/s41598-017-05888-8>.
 - [27] L. Wu, Y. Wang, Y. Xia, F. Tian, F. Gao, T. Qin, J. Lai, and T.-Y. Liu, "Depth growing for neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 5558–5563. [Online]. Available: <https://aclanthology.org/P19-1558/>.
 - [28] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser, "Universal transformers," *arXiv preprint arXiv:1807.03819*, 2018, pp. 1–23. [Online]. Available: <https://arxiv.org/abs/1807.03819>.
 - [29] Y. Xiao, L. Wu, J. Guo, J. Li, M. Zhang, and T. Qin, "A survey on non-autoregressive generation for neural machine translation and beyond," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11407–11427, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10129160>.
 - [30] J. Lee, E. Mansimov, and K. Cho, "Deterministic non-autoregressive neural sequence modeling by iterative refinement," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1173–1182. [Online]. Available: <https://aclanthology.org/D18-1149/>.
 - [31] M. Ghazvininejad, O. Levy, Y. Liu, and L. Zettlemoyer, "Mask-predict: Parallel decoding of conditional masked language models," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 6112–6121. [Online]. Available: <https://aclanthology.org/D19-1633/>.
 - [32] A. Bapna, M. Chen, O. Firat, Y. Cao, and Y. Wu, "Training deeper neural machine translation models with transparent attention," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 3028–3033. [Online]. Available: <https://aclanthology.org/D18-1338/>.
 - [33] B. Zhang, I. Titov, and R. Sennrich, "Improving deep transformer with depth-scaled initialization and merged attention," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 898–909. [Online]. Available: <https://aclanthology.org/D19-1083/>.
 - [34] N. Shahin and L. Ismail, "From rule-based models to deep learning transformer architectures for natural language processing and sign language translation systems: survey, taxonomy and performance evaluation," *Artificial Intelligence Review*, vol. 57, no. 10, pp. 1–51, 2024. [Online]. Available: <https://link.springer.com/article/10.1007/s10462-024-10895-z>.
 - [35] L. Kaiser, A. N. Gomez, and F. Chollet, "Depthwise separable convolutions for neural machine translation," *arXiv preprint arXiv:1706.03059*, 2017, pp. 1–10. [Online]. Available: <https://arxiv.org/abs/1706.03059>.
 - [36] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, and M. Hughes, "The best of both worlds: Combining recent advances in neural machine translation," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2018, pp. 76–86. [Online]. Available: <https://aclanthology.org/P18-1008/>.
 - [37] F. Meng and J. Zhang, "DTMT: A novel deep transition architecture for neural machine translation," in *Proceedings of AAAI*, 2019, pp. 224–231. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/3789>.
 - [38] B. Zhang, D. Xiong, and J. Su, "Neural machine translation with deep attention," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 1, pp. 154–163, 2020. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/8493282>.
 - [39] Y. Li, J. Li, and M. Zhang, "Deep transformer modeling via grouping skip connection for neural machine translation," *Knowledge-Based Systems*, vol. 234, p. 107556, 2021. [Online]. Available: <https://doi.org/10.1016/j.knsys.2021.107556>.
 - [40] X. Wei, H. Yu, Y. Hu, Y. Zhang, R. Weng, and W. Luo, "Multiscale collaborative deep models for neural machine translation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 414–426. [Online]. Available: <https://aclanthology.org/2020.acl-main.40/>.
 - [41] B. Li, Z. Wang, H. Liu, Y. Jiang, Q. Du, T. Xiao, H. Wang, and J. Zhu, "Shallow-to-deep training for neural machine translation," in *Proceedings of EMNLP*, 2020, pp. 995–1005. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.72/>.
 - [42] X. Liu, K. Duh, L. Liu, and J. Gao, "Very deep transformers for neural machine translation," in *arXiv preprint arXiv:2008.07772*,

- 2020, pp. 1–7. [Online]. Available: <https://arxiv.org/abs/2008.07772>.
- [43] C. Qin, L. Chen, Z. Cai, M. Liu, and L. Jin, "Long short-term memory with activation on gradient," *Neural Networks*, vol. 164, pp. 135–145, 2023. [Online]. Available: <https://doi.org/10.1016/j.neunet.2023.04.026>.
- [44] C. L. Galimberti, L. Furieri, L. Xu, and G. Ferrari-Trecate, "Hamiltonian deep neural networks guaranteeing nonvanishing gradients by design," *IEEE Transactions on Automatic Control*, vol. 68, no. 5, pp. 3155–3162, 2023. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/10025364>.
- [45] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *Proceedings of CVPR*, 2015, pp. 3367–3375. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2015/html/Liang_Recurrent_Convolutional_Neural_2015_CVPR_paper.html.
- [46] X. Zhang, C. Zhong, J. Zhang, T. Wang, and W. W. Y. Ng, "Robust recurrent neural networks for time series forecasting," *Neurocomputing*, vol. 526, pp. 143–157, 2023. [Online]. Available: <https://doi.org/10.1016/j.neucom.2023.01.037>.
- [47] Z. Tu, Z. Lu, Y. Liu, X. Liu, and H. Li, "Modeling coverage for neural machine translation," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016, pp. 76–85. [Online]. Available: <https://aclanthology.org/P16-1008.pdf>.
- [48] D. R. So, C. Liang, and Q. V. Le, "The evolved transformer," in *Proceedings of the 36th International Conference on Machine Learning*, 2019, pp. 5877–5886. [Online]. Available: <https://proceedings.mlr.press/v97/so19a>.
- [49] X. Li, A. C. Stickland, Y. Tang, and X. Kong, "Deep transformers with latent depth," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1736–1746, 2020. [Online]. Available: <https://proceedings.neurips.cc/paper/2020/hash/1325cdac3b6f0f91a1b629307bf2d498-Abstract.html>.
- [50] M. Ott, S. Edunov, D. Grangier, and M. Auli, "Scaling neural machine translation," in *Proceedings of the Third Conference on Machine Translation*, 2018, pp. 1–9. [Online]. Available: <https://aclanthology.org/W18-6301/>.
- [51] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: Open-source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, 2007, pp. 177–180. [Online]. Available: <https://aclanthology.org/P07-2045/>.
- [52] M. Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation*, 2018, pp. 186–191. [Online]. Available: <https://aclanthology.org/W18-6319/>.
- [53] A. Fan, E. Grave, and A. Joulin, "Reducing transformer depth on demand with structured dropout," *arXiv preprint arXiv:1909.11556*, 2019. [Online]. Available: <https://arxiv.org/abs/1909.11556>.
- [54] A. Conneau, G. Kruszewski, G. Lample, L. Barrault, and M. Baroni, "What you can cram into a single \$&! #* vector: Probing sentence embeddings for linguistic properties," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018, pp. 2126–2136. [Online]. Available: <https://aclanthology.org/P18-1198/>.
- [55] X. Wang, Z. Tu, L. Wang, and S. Shi, "Exploiting sentential context for neural machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 6197–6203. [Online]. Available: <https://aclanthology.org/P19-1624/>.
- [56] M. Yang, R. Wang, K. Chen, X. Wang, T. Zhao, and M. Zhang, "A novel sentence-level agreement architecture for neural machine translation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2585–2597, 2020. [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/9185042>.
- [57] Q. Wang, B. Li, T. Xiao, J. Zhu, C. Li, D. F. Wong, and L. S. Chao, "Learning deep transformer models for machine translation," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 1810–1822. [Online]. Available: <https://aclanthology.org/P19-1176/>.
- [58] J. Li, D. Xiong, Z. Tu, M. Zhu, M. Zhang, and G. Zhou, "Modeling source syntax for neural machine translation," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017, pp. 688–697. [Online]. Available: <https://aclanthology.org/P17-1064/>.
- [59] Y. Li, J. Li, and M. Zhang, "Adaptive weighting for neural machine translation," in *Proceedings of the 27th International Conference on Computational Linguistics*, 2018, pp. 3038–3048. [Online]. Available: <https://aclanthology.org/C18-1257/>.
- [60] Y. Li, J. Li, M. Zhang, Y. Li, and P. Zou, "Improving neural machine translation with linear interpolation of a short-path unit," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 19, pp. 1–16, 2020. [Online]. Available: <https://dl.acm.org/doi/abs/10.1145/3377851>.
- [61] X. Lyu, J. Li, Z. Gong, and M. Zhang, "Encouraging lexical translation consistency for document-level neural machine translation," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 3265–3277. [Online]. Available: <https://aclanthology.org/2021.emnlp-main.262/>.
- [62] R. Sun, H. Jin, and X. Wan, "Document-level text simplification: Dataset, criteria and baseline," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 7997–8013. [Online]. Available: <https://aclanthology.org/2021.emnlp-main.630/>.

Author Biography



Jinxi Zhang is an associate professor and Ph.D. at the School of Business and Media, Lanzhou University of Finance and Economics. His research interests include natural language processing, speech engineering, and experimental phonetics.
ORCID: 0009-0005-3632-2655, Email: 582361484@qq.com



Chunling Li is a Ph.D. student at the School of Statistics and Data Science, Lanzhou University of Finance and Economics. Her research interests include machine learning and data mining.
ORCID: 0009-0007-4271-5214, Email: lichunling@lzufe.edu.cn

Appendix A. Ablation Study of the Fusion Function in Eq. (4)

Table A1. Ablation study of fusion functions in Eq. (4) on WMT16 En-De (Transformer-base setting).

Fusion Function	Extra Parameters	BLEU	SBLEU
Addition	0	29.4	28.3
Mean Pooling	0	29.2	28.1
Gating Mechanism	+0.3M	29.3	28.2

As shown in Table A1, the translation quality achieved by these three fusion functions is similar, with relatively small differences. Among them, simple addition provides the best overall balance between effectiveness and simplicity. The performance of average pooling is slightly inferior, while the gating mechanism introduces additional fusion parameters but does not bring significant performance advantages. Therefore, we adopt simple addition as the default fusion function in Eq. (4).