THOR-MoE: Hierarchical Task-Guided and Context-Responsive Routing for Neural Machine Translation

Yunlong Liang, Fandong Meng*, Jie Zhou
Pattern Recognition Center, WeChat AI, Tencent Inc
{yunlonliang,fandongmeng,withtomzhou}@tencent.com

Abstract

The sparse Mixture-of-Experts (MoE) has achieved significant progress for neural machine translation (NMT). However, there exist two limitations in current MoE solutions which may lead to sub-optimal performance: 1) they directly use the task knowledge of NMT into MoE (*e.g.*, domain/linguistics-specific knowledge), which are generally unavailable at practical application and neglect the naturally grouped domain/linguistic properties; 2) the expert selection only depends on the localized token representation without considering the context, which fully grasps the state of each token in a global view. To address the above limitations, we propose THOR-MoE via arming the MoE with hierarchical task-guided and context-responsive routing policies. Specifically, it 1) firstly predicts the domain/language label and then extracts mixed domain/language representation to allocate task-level experts in a hierarchical manner; 2) injects the context information to enhance the token routing from the pre-selected task-level experts set, which can help each token to be accurately routed to more specialized and suitable experts. Extensive experiments on multi-domain translation and multilingual translation benchmarks with different architectures consistently demonstrate the superior performance of THOR-MoE. Additionally, the THOR-MoE operates as a plug-and-play module compatible with existing Top-k (Shazeer et al., 2017a) and Top-p (Huang et al., 2024) routing schemes, ensuring broad applicability across diverse MoE architectures. For instance, compared with vanilla Top-p (Huang et al., 2024) routing, the context-aware manner can achieve an average improvement of 0.75 BLEU with less than 22% activated parameters on multi-domain translation tasks.

* Corresponding author.

1 Introduction

The rapid advancement of neural machine translation (NMT; Zhang et al. (2020); Fan et al. (2020)) has been significantly propelled by sparse Mixture-of-Experts (MoE) architectures (Shazeer et al., 2017a; Lepikhin et al., 2021a; Fedus et al., 2022), which increase model capacity while maintaining computational efficiency through conditional computation. In the literature, existing work can be roughly classified into two categories: (1) some studies mainly focus on how to effectively incorporate task information (*e.g.,* domain/linguistics-specific knowledge) into MoE models designing task-specific modules (Pham et al., 2023; Jiang et al., 2024); (2) another line of work concentrates on how to improve the training and inference efficiency via reducing the activated experts in MoE (Jawahar et al., 2023a; Elbayad et al., 2023; Wang et al., 2024; Zeng et al., 2024).

Specifically, in (1), Kudugunta et al. (2021) routes input examples to different experts based on translation language representations. Li et al. (2023c) explores multiple language-group-specific routers to incorporate language group knowledge into models. Zhao et al. (2024b) takes one step further by designing hierarchical language-guided token routing. Although intuitive and effective, they heavily rely on explicit task-specific knowledge (*e.g.*, domain or language labels), which is often unavailable in real-world scenarios. This reliance not only restricts generalization but also overlooks the intrinsic hierarchical grouping of domain and linguistic properties inherent in multilingual or multi-domain translation tasks. In (2), Zhao et al. (2024a) distills the knowledge of unselected experts to the selected one and thus reduces the number of experts without impacting performance; Li et al. (2023a) and Huang et al. (2024) decrease the number of experts by designing a threshold based on expert probability distribution; Zhao et al.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 21433–21445

(2024b) adapts the number of experts by the difficulty of language. Generally, the global context grasp the overall situation and might know whether each token is difficult or not (Gloeckle et al., 2024). However, previous routing strategies select experts solely based on localized token-level representations, neglecting the broader contextual dependencies that govern token interactions. This myopic view limits the model's ability to allocate experts optimally in a globally coherent manner.

In this work, to address the above issues, we propose a innovative hierarchical context-responsive routing (THOR-MoE) framework for neural machine translation. Specifically, the THOR-MoE first predicts which domain/language the input belongs to and then extracts the mixed task[1] representation to allocates experts in a hierarchical manner. It is not only flexible in practice but also effectively infuses the domain knowledge into the task-level routing via encouraging experts to specialize in certain domains/languages. Furthermore, before each token routing, the THOR-MoE injects the context information to help accurately assign the experts from the pre-selected task-level experts set for each token in a global perspective. This enables the model to adaptively capture evolving contextual dependencies, such as long-range syntactic structures or discourse coherence. Crucially, the THOR-MoE operates as a plug-and-play module compatible with existing Top-k (Shazeer et al., 2017a) and Top-p (Huang et al., 2024) routing schemes, ensuring broad applicability across diverse MoE architectures.

We validate our proposed THOR-MoE framework on the commonly-used multi-domain NMT benchmark (Aharoni and Goldberg, 2020) and multilingual NMT benchmark (Zhang et al., 2020), which contains 5 domains and 16 languages, respectively. Extensive experiments show that the hierarchical context-responsive routing is pivotal for unlocking the full potential of MoE systems.

In summary, our main contributions are:

- We propose two components to expand MoE: (i) a hierarchical design to enable language- and domain-specific expert routing; (ii) enabling the use of context information during expert selection, which is important for translation.
- Extensive experiments on both multi-domain and multilingual translation consistently show the effectiveness and generalization of THOR-MoE. For example, our model achieves consistent improvements of an average improvement of 0.75 BLEU with less than 22% activated parameters over vanilla Top-p routing in multi-domain translation tasks.

[1]Note that the task generally denotes the domain/language-specific knowledge in this work.

2 Background

2.1 Neural Machine Translation

Given an input sentence in the source language $X=\{x_i\}_{i=1}^{|X|}$, the goal of the model is to produce its translation in the target language $Y=\{y_i\}_{i=1}^{|Y|}$. The conditional distribution of the model is:

$$\mathcal{L}_{\text{NMT}} = -\sum_{t=1}^{|Y|} \log(p(y_t|X, y_{1:t-1})), \quad (1)$$

where $y_{1:t-1}$ is the partial translation.

2.2 Sparse Mixture-of-Experts

The vanilla MoE model can be seen the variant of Transformer model via replacing the feed-forward (FFN) sub-block of the transformer block with MoE layer, in which per token selects fixed number of experts (Lepikhin et al., 2021b). In each MoE layer, there are N experts, denoted as $\mathcal{E} = \{e_1, e_2, \ldots, e_N\}$. An input $\mathbf{x}$ is dispatched to these experts, and the output of the MoE layer is computed as the weighted average of the outputs from the experts:

$$\text{MoE}(\mathbf{x}) = \sum_{i=1}^{N} g_i(\mathbf{x}) * e_i(\mathbf{x}), \quad (2)$$

where $g_*(\mathbf{x})$ is computed by a router that predicts the contribution of each expert to the final output. Given the computing efficiency, the MoE assigns each token to limited experts (*e.g.*, 1 or 2).

To derive $g_*(\mathbf{x})$, we generally compute the probability $\mathbf{P}$ of each expert being selected for the input $\mathbf{x}$ as follows:

$$\mathbf{P} = \text{Softmax}(\mathbf{W_r} \cdot \mathbf{x}^T), \quad (3)$$

where $\mathbf{W_r} \in N \times d$ serves as a learnable parameter and d denotes the dimension of the input $\mathbf{x}$. The vector $\mathbf{P}$, with a size of N, encapsulates the probabilities associated with the selection of each expert. And P_i indicates the likelihood of choosing the i^{th} expert e_i to calculate the input $\mathbf{x}$.

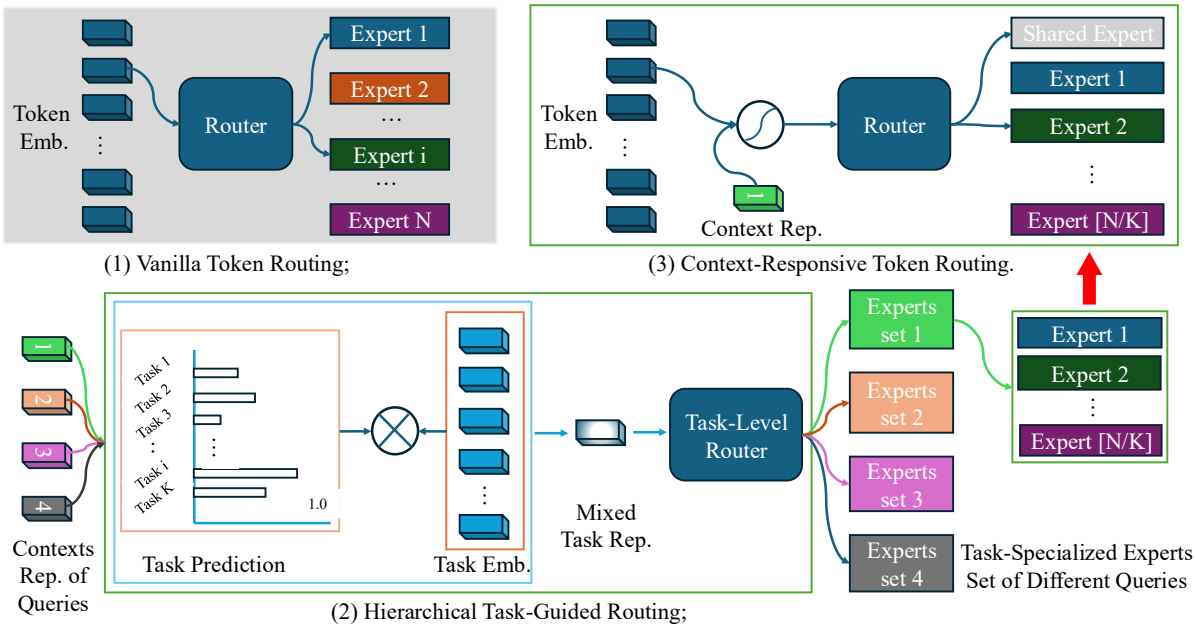

Figure 1: The overview of the proposed THOR-MoE. 1) vanilla token routing; 2) hierarchical task-guided routing; 3) context-responsive token routing. The 'Emb.' and 'Rep.' denotes the embedding and representation, respectively. The hierarchical manner denotes that the task-guided routing firstly selects different task-specific experts sets for different queries (*e.g.*, $\mathcal{E}_t$ for query 1). Then the context-responsive token routing assigns experts from $\mathcal{S}^t$ for each token in query 1.

2.2.1 Top-k Routing

Top-k routing (Shazeer et al., 2017b; Lepikhin et al., 2021b; Zuo et al., 2022) entails the selection of the top k experts, which correspond to the k highest probabilities within the vector $\mathbf{P}$. Then, the probabilities of the chosen experts are re-scaled through normalization, while the probabilities of the non-selected experts are set to zero, signifying their deactivation. The resultant computation for $g_*(\mathbf{x})$ proceeds as below:

$$g_i(\mathbf{x}) = \begin{cases} \frac{P_i}{\sum_{j \in \text{TopK}(\mathbf{P})} P_j}, & i \in \text{TopK}(\mathbf{P}) \\ 0, & i \notin \text{TopK}(\mathbf{P}) \end{cases} \quad (4)$$

where $\text{TopK}(\mathbf{P})$ returns the indices of the k largest elements in $\mathbf{P}$.

2.2.2 Top-p Routing

Top-p routing is initially proposed by (Li et al., 2023a; Huang et al., 2024), which aims to overcome the drawback of Top-k activating the fixed number of experts overlooking the variability in difficulty across different inputs. We refer to readers to Huang et al. (2024) for more details.

Formally, Top-p first sorts the elements in $\mathbf{P}$ from highest to lowest, returning in a sorted index list I. Subsequently, we identify the minimal subset of experts for which the aggregate probability surpasses the threshold p. The count of selected experts, denoted as t, is then determined by:

$$t = \underset{k \in \{1...,N\}}{\arg\min} \sum_{j<=k} P_{i,j} \geq p, \quad (5)$$

where p is the threshold that controls how confident the model should be when stopping adding more experts. p is a hyper-parameter whose range is from 0 to 1. The higher the p is, the more experts will be activated.

In dynamic routing mechanism, the calculation of $g_*(\mathbf{x})$ is:

$$g_i(\mathbf{x}) = \begin{cases} P_i & e_i \in S \\ 0, & e_i \notin S \end{cases} \quad (6)$$

where S is the set of selected experts controlled by t in Equation 5:

$$S = \{e_{I_1}, e_{I_2} ... e_{I_t}\}. \quad (7)$$

3 THOR-MoE

In this section, we introduce the proposed hierarchical context-responsive routing framework as shown in Figure 1, including two components: *hierarchical task-guided routing* (§ 3.1) and *context-responsive token routing* (§ 3.2). Finally, we describe the *training objectives* (§ 3.3) in detail.

3.1 Hierarchical Task-Guided Routing

Our hierarchical task-guided routing consists of three stages, as shown in 2) of 1. The first stage is task prediction, which aims to automatically obtain the task knowledge by predicting which domain/language the input belongs to. The second stage is to inject the mixed task representation to allocates experts in a hierarchical task-guided manner (third stage).

Task Predictor. During real testing for a query, it is hard to reach its domain or language category. Therefore, to automatically obtain the domain knowledge or language knowledge, we add a special token [CLS] in front of the input. Then, it represents the global sentence information and thus can effectively reflect which domain/language the input belongs to (Devlin et al., 2019). Finally, we transform the representation $\mathbf{H}^{cls}$ into a fixed-size vector through $\mathrm{Maxpooling}$ and apply a fully-connected layer to predict suitable labels:

$$\begin{aligned} \mathbf{H}^{max} &= \mathrm{Maxpooling}(\mathbf{H}^{cls}),\ \mathbf{H}^{max} \in \mathbb{R}^{d}, \\ \mathcal{P}^{t} &= \mathrm{Softmax}(\mathbf{W}^{p}\mathbf{H}^{max}), \end{aligned} \tag{8}$$

where $\mathbf{W}^{p} \in \mathbb{R}^{K\times d}$ is trainable weight and K denotes the number of domains or languages and d is the model dimension.

Mixed Task Representation. Theoretically, we can directly take the predicted task label to extract corresponding task representation. However, in some cases in practice, it is hard to judge since some sentences belongs to multiple domains. In language cases, there are some code-mixed sentences, which is also confused to decide. Therefore, directly use the given label in existing work or the predicted label cannot accurately help model select the specialized experts for such cases and thus limits their potential, as shown in § 6.2. To address this issue, we use the mixed task representation as follows:

$$\mathbf{E}_{p} = \sum(\mathcal{P}^{t} \cdot \mathbf{EMB1}),\ \mathbf{E}_{p} \in \mathbb{R}^{d}, \tag{9}$$

where $\mathbf{E}_{p}$ denotes the mixed task representation generated by the weighted sum of the task distribution $\mathcal{P}$ (predicted by the task predictor as Eq. 8) and the task parameter matrix $\mathbf{EMB1}$. The dimension of $\mathbf{EMB1}$ is num (number of tasks/language groups) * d (embedding dimension), which is randomly initialized and then optimized with the training objective.

Hierarchical Task-Guided Routing. With the mixed task representation, we design a task router g^{t} at the task level. In each MoE layer, g^{t} takes the mixed task representation $\mathbf{E}_{p}$ as input and outputs a task-dependent expert vector. Then, based on it, we use TopK function to select task-specific candidate experts as $\mathcal{S}^{t}$ from all experts. Finally, based on the selected task-specific candidate experts $\mathcal{S}_{t}$, we can conduct general token routing at the token level with Top-k or Top-p routing as described in § 2.2.1 and § 2.2.2, respectively. The final routing policy can be written as $\sum_{i}^{|\mathcal{S}^{t}|}\sum_{j}^{|\mathcal{E}|} g_{i}^{t}\cdot g_{j}\cdot e_{i,j}(\mathbf{x})$.

However, current expert selection at the token level only depends on the current token representation without considering the complete word meaning and global context. Generally, the global context grasp the overall situation and might know which token is difficult or not. Thus, the context can schedules the token to route to suitable experts in a global view. To this end, we propose context-responsive experts selection at the token level.

3.2 Context-Responsive Routing

As shown in 3) of 1, before the experts selection at the token level, we explicitly inject the context into the token representation. Since it is hard to determine for a code-mixed input or a query that can belongs to several domains, which thus limits its application in practice. Therefore, we use the averaged context representation as the context representation. $\mathbf{H}_{ctx} = \frac{1}{|Y_{1:t-1}|}\sum_{t=1}^{|Y_{1:t-1}|}\mathbf{h}_{t}^{L}$ where $\mathbf{h}$ is the hidden state of the t-th token at the L layer. At each decoding step, the context used for each input token is the prefix of the generated next token. That is, the context is dynamically updated with decoding.

To effectively incorporate the context representation into the routing process, we design a gate to dynamically control the contribution of these information:

$$\begin{aligned} \mathbf{x}_{i} &= g\odot\mathbf{x}_{i} + (1-g)\odot\mathbf{H}_{ctx}, \\ g &= \sigma([\mathbf{x}_{i};\mathbf{H}_{ctx}]\mathbf{W}^{g} + \mathbf{b}^{g}), \end{aligned}$$

where $\mathbf{W}^{g}$ and $\mathbf{b}^{g}$ are the trainable parameters.

In this manner, we can inject the context information into each token to accurately assign specialized experts to suitable tokens and unlock the potential of the MoE.

3.3 Training Objectives

Our training objectives consist of four parts in Top-k routing (translation loss $\mathcal{L}_{\text{NMT}}$, task prediction loss $\mathcal{L}_{\text{tp}}$, load balance loss at the task level $\mathcal{L}_{bd}$

and token level $\mathcal{L}_{bt}$) and five parts in Top-p routing ($\mathcal{L}_{\text{NMT}}$, $\mathcal{L}_{\text{tp}}$, $\mathcal{L}_{bd}$, $\mathcal{L}_{bt}$ and dynamic routing loss $\mathcal{L}_{d}$ in Top-p).

Task Prediction Loss. The task prediction loss is defined as:

$$\mathcal{L}_{tp} = \min(-\log(\mathcal{P}^t[t_g])), \tag{10}$$

where t_g is the golden class label, $\mathcal{P}^t$ is calculated as in Eq. 8.

Load Balance Loss at the Task Level. To fully encourage the experts to specialize in certain tasks, we propose a task-guided routing loss. It aims to achieve the number of tasks processed by different experts to be roughly the same. Therefore, the task-guided routing loss is defined as:

$$\mathcal{L}_{bd} = N * \sum_{i=1}^{N} F_i^t * Q_i^t, \tag{11}$$

where F_i^t represents the proportion of tasks selecting expert e_i, and Q_i^t denotes the proportion of the router's probability allocated to expert e_i. For K tasks, F_i^t and Q_i^t are defined as: $F_i^t = \frac{1}{K}\sum_{j=1}^{K} 1\{e_i \in \mathcal{S}^{t,j}\}$ and $Q_i^t = \frac{1}{K}\sum_{j=1}^{K} P_i^j$ where $\mathcal{S}^{t,j}$ is the set of activated experts for task j, which is calculated by Equation 7, and P^j is the probability of selecting each experts for task j, calculated by Equation 3.

Load Balance Loss at the Token Level. This loss is the similar to the vanilla load balance loss with a slight difference that the experts candidates is pre-selected by the hierarchical task-guided routing. Formally, the load balance loss at the token level is defined as:

$$\mathcal{L}_{bt} = |\mathcal{S}^t| * \sum_{k=1}^{|\mathcal{S}^t|} F_k^b * Q_k^b, \tag{12}$$

where F_k^b represents the proportion of tokens selecting expert e_k, and Q_k^b denotes the proportion of the router's probability allocated to expert e_k. For a sequence comprising M tokens, F_k^b and Q_k^b are defined as: $F_k^b = \frac{1}{M}\sum_{\ell=1}^{M} 1\{e_k \in \mathcal{S}^{t,\ell}\}$ and $Q_k = \frac{1}{M}\sum_{\ell=1}^{M} P_k^\ell$ where $\mathcal{S}^{t,\ell}$ is the set of activated experts for token ℓ from pre-selected experts set $S^{d,\ell}$, which is calculated by Equation 7, and P^ℓ is the probability of selecting each experts for token ℓ, calculated by Equation 3.

Dynamic Routing Loss in Top-p. The dynamic routing loss aims to prevent dynamic routing from using too many parameters to cheat and losing its ability to selectively choose experts. Therefore, Huang et al. (2024) introduce a constraint on $\mathbf{P}$ and minimize the entropy of the distribution $\mathbf{P}$, ensuring that every token can focus on as less specific experts as possible, which is formalized as:

$$\mathcal{L}_{topp} = -\sum_{i=1}^{N} P_i * log(P_i). \tag{13}$$

Final Loss. Our approach can be flexibly applied in Top-k and Top-p routing strategies. Therefore, we have two final loss functions for Top-k and Top-p, respectively. It is a combination of the translation loss, task prediction loss, load balance loss at the task level and at the token level, and dynamic loss:

$$\mathcal{J}_{topk} = \mathcal{L}_{\text{NMT}} + \alpha\mathcal{L}_{dp} + \beta\mathcal{L}_{bd} + \gamma\mathcal{L}_{bp}, \tag{14}$$

$$\mathcal{J}_{topp} = \mathcal{J}_{topk} + \delta\mathcal{L}_{topp}, \tag{15}$$

where α, β, γ, and δ are hyper-parameters to adjust the contribution among these loss functions, respectively.

4 Experiments

4.1 Datasets and Metric

Datasets. We use the multi-domain translation dataset proposed by Koehn and Knowles (2017). The dataset mainly covers five diverse domains: IT, Koran, Law, Medical, and Subtitles, which are available in OPUS (Aulamo and Tiedemann, 2019). Following previous work (Gu et al., 2022; Liang et al., 2024), we use the new data splitting released by Aharoni and Goldberg (2020), and perform German to English translation (De→En). We use the OPUS-16 for multilingual translation following (Zhao et al., 2024b). The OPUS-16 comes from OPUS-100 (Zhang et al., 2020), which includes 16 languages (8 high resource (> 0.9M), 4 medium resource, and 4 low resource (< 0.1M)). Please refer to Tab. 8 of § 8 for detailed data statistics.

Metric. For a fair comparison, we follow previous work (Gu et al., 2022; Zhao et al., 2024b) and adopt the 4-gram case-sensitive BLEU with the SacreBLEU tool[2] (Post, 2018) and report the statistical significance test (Koehn, 2004). For

[2]BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.4.13

	Models	IT	Koran	Medical	Law	Subtitles	Avg.
Dense	SFT-3B	40.65	20.4	51.4	54.8	28.33	39.12
MoE	Trim-MoE (Top-1)	40.03	19.55	51.80	55.83	25.50	38.54
	Trim-MoE (Top-2)	<u>45.10</u>	<u>22.68</u>	51.84	57.12	<u>29.02</u>	41.15
	Trim-MoE (Top-p, p=0.5)	39.39	19.21	<u>55.67</u>	<u>60.18</u>	**29.21**	40.73
	THOR-MoE (Top-1)	43.96‡	21.93‡	52.52†	56.25	28.41‡	40.61‡
	THOR-MoE (Top-2)	**46.00**†	**23.35**	**55.79**‡	**61.06**‡	28.23	**42.89**‡
	THOR-MoE (Top-p, p=0.5)	44.63‡	22.53‡	53.58	58.65	27.99	<u>41.48</u>†

Table 1: BLEU score on multi-domain translation benchmarks with decoder-only architecture. "†" and "‡" denote that statistically significant better than the best result of the counterpart (*e.g.*, THOR-MoE (Top-2) vs. Trim-MoE (Top-2)) with t-test $p < 0.05$ and $p < 0.01$ hereinafter, respectively. The best and second best results are **bold** and <u>underlined</u>, respectively.

multilingual translation, the scores encompass the average ratings across all language pairs, such as English→Any (En→XX), and Any→English (XX→En) on the OPUS-16 dataset.

4.2 Implementation Details

In multi-domain translation, we use decoder-only Transformer architecture. Specifically, we use Qwen1.5-MoE-A2.7B (Team, 2024a). It has 14.3B parameters (60 non shared experts + 4 shared experts) in total and 2.7B activated parameters (4 non shared experts + 4 shared experts) during runtime. Due to limited GPU resource, we used a trimmed version, which has 3.5B parameters (8 non shared experts + 4 shared experts) in total and 2.3B activated parameters (2 non shared experts + 4 shared experts). To keep its capacity, we initialize the trimmed model with Qwen1.5-MoE-A2.7B and denote is as Trim-MoE. During training, we use Llama-Factory (Zheng et al., 2024) to instruct-tune LLMs. All LLMs are tuned on an 8×NVIDIA A100 GPUs (40G) with 1e-5 learning rate. We set gradient accumulation to 16 and batch size to 1, which gives us 2*8*16*1 batch in total. We use the DeepSpeed optimization (Rasley et al., 2020), and set ZeRO-3 optimization. Following Qin et al. (2024), we set the number of training epochs to 3. We set hyper-parameters α, β, γ, and δ to 1e-2, 1e-2, 1e-2, and 1e-4, which are determined by a grid search.

In multilingual translation, we follow (Zhao et al., 2024b) and compare our method with the Transformer-Base model (as Dense) and its MoE variants that have 6 encoder and decoder layers, 32 experts. The input and hidden dimensions of all feed-forward networks are 512 and 2048. We set the training processes to 35K iterations with a learning rate of 5e-4, which follows the inverse square root with 4,000 warm-up steps. To keep balanced training, we use temperature-based data sampling strategy with temperature 1.5. We set hyper-parameters α, β, γ, and δ to 1e-2, 5e-2, 5e-2, and 1e-4, which are determined by a grid search.

4.3 Comparison Models

Our comparison models mainly include two types: **Decoder-only** and **Encoder-Decoder** based.

Decoder-only. We fine-tune two dense models based on Qwen2.5-3B (Team, 2024b), which have similar parameters with the activated parameter and denoted as SFT-3B. Besides, based on the Trim-MoE-A2.3B model, we also fine-tuned three MoE models with Top-1, Top-2, and Top-p routing strategy.

Encoder-Decoder. There are three types of comparison models: vanilla dense model, vanilla MoE model, and enhanced MoE models via language knowledge. We use the Transformer-Base model as Dense model. For vanilla MoE model, the Switch Transformer (Fedus et al., 2022) with a top-1 token-based routing (as ST-MoE) and GShard (Lepikhin et al., 2021a) with a top-2 token-based routing (as GS-MoE) are adopted. For Language-specific MoE models, the LS-MoE with fixed routing inspired by (Pires et al., 2023), assigning 2 non-overlapping experts for tokens according to their source language in the encoder and target language in the decoder; Hybrid-MoE (Kudugunta et al., 2021), with a top-2 token routing in the encoder and a top-2 target language routing in the decoder side; Residual-MoE (Elbayad et al., 2023; Rajbhandari et al., 2022; Zhang et al., 2021) that augments each MoE layer with a shared feed-forward network through a binary gate function; Lingual-

MoE (Zhao et al., 2024b) stands for an MoE model with linguistic-guided routing and dynamic expert allocation, where the first-level language router selects the top 8 experts and the second-level token router activates the dynamical number of experts.

5 Main Results

Table 1 shows main results on the multi-domain translation benchmark with decoder-only architecture. Table 2 presents main results on the multilingual translation with encoder-decoder architecture. For a fair comparison, all models are trained and assessed on the OPUS-16 dataset with Transformer-Base as the backbone architecture.

5.1 Results on Multi-Domain Translation

Table 1 shows that the Trim-MoE-based models generally surpass the dense one. For example, the Trim-MoE (Top-2) outperforms SFT-3B model by average 2 BLEU scores where the activated parameters is less than dense model (2.3B vs. 3B), showing the effectiveness of MoE model. Furthermore, the proposed THOR-MoE significantly and consistently surpasses the counterpart of Trim-MoE-based one. For instance, the THOR-MoE (Top-2) outperforms the Trim-MoE (Top-2) by averaged 1.74 BLEU scores. This clear advantage confirms the superiority of incorporating domain knowledge and context knowledge into the routing. What's important, the proposed approach is compatible with the Top-k (Shazeer et al., 2017a) and Top-p (Huang et al., 2024) routing strategies, validating the generalization of THOR-MoE, which can be a plug-and-play module. Besides, we find that the scores on the same domain (*e.g.*, Law) change largely. The reason may be that the data distribution is unbalanced. We also list the results in terms of COMET (Rei et al., 2020) score in Table 3 and we can conclude the similar findings.

5.2 Results on Multilingual Translation

Table 2 summarizes the results and we can conclude several observations:

THOR-MoE vs. Dense and Vanilla MoE Baselines. The THOR-MoE substantially surpasses the Dense and vanilla MoE multilingual translation baselines with a large margin. Specifically, compared with Dense and ST-MoE, the THOR-MoE (Top-p) achieves {5.39%, 3.99%, 4.70%} and 2.46%, 0.55%, 1.51% averaged improvement in terms of BLEU score for Avg1., Avg2., and All Avg., respectively. This significant margin demonstrates the effectiveness of hierarchical language-guided routing and context-responsive routing.

THOR-MoE vs. Language-Guided MoE models. The THOR-MoE also consistently outperforms language-guided MoE models, including LS-MoE, hybrid-MoE, Residual-MoE, and Lingual-MoE. It shows again that the superiority of hierarchical language-guided routing and context-responsive routing. In detail, the Lingual-MoE performs the highest in baselines and it also employs a hierarchical language-group-guided and dynamical routing, which introduces the hard language id embedding and fails to consider code-mixed cases and that the token in different language can be the same (*e.g.*, 'Internet' in German and English). In contrast, THOR-MoE applies a mixed language representation where the language ids are predicted, which comprehensively incorporates the languages knowledge. Furthermore, the dynamical routing in Lingual-MoE does not incorporate the context information. The context generally knows which token is difficult or not in a global view. The THOR-MoE fully considers the above issues and obtains better results.

6 Analysis

6.1 Ablation Study

We conduct ablation studies to investigate how well hierarchical context-responsive routing of THOR-MoE works. We conclude two findings from the results in Tab. 4.

(1) "*w/o* hierarchical task-guided routing": *i.e.*, without using any task-related routing and decaying to vanilla manner (select experts from full set of experts), the model performance greatly degrades on both translation tasks. It shows the necessity of using task-guided routing in a hierarchical manner.

(2) "*w/o* context-responsive routing": the model performance becomes worse on both tasks when removing context. This shows that our context-responsive routing indeed can enhance the routing effectiveness and guarantee the token can be assigned to suitable and specialized experts with the indication of context, which thus benefits the model performance on both translation tasks.

6.2 Comparison among Different Task Representations

In this section, we aim to investigate the impact of task representation in different manners. Table 5

	Models	En→XX				XX→En				Avg.
		high	medium	low	Avg1.	high	medium	low	Avg2.	
Dense	Transformer-base	25.37	39.12	14.78	26.16	28.81	39.24	24.21	30.27	28.21
MoE	GS-MoE	25.55	40.71	18.97	27.70	28.77	41.70	29.25	32.12	29.91
	ST-MoE	26.55	43.04	20.23	29.09	29.94	44.00	30.94	33.71	31.40
	LS-MoE	20.06	37.31	11.72	22.29	23.69	42.99	32.01	30.60	26.44
	Hybird-MoE	24.56	31.70	13.38	23.55	29.59	41.74	27.81	32.18	27.85
	Residual-MoE	26.52	43.49	21.58	29.53	29.84	43.94	31.25	33.72	31.62
	Lingual-MoE	27.11	46.24	23.34	30.95	29.87	43.49	32.02	33.81	32.38
	THOR-MoE (Top-1)	27.21	46.37	23.88	31.17	29.68	44.01	32.58	33.99	32.57
	THOR-MoE (Top-2)	**27.88**†	**47.29**†	**24.85**‡	**31.98**†	**30.38**†	**44.52**†	**33.26**†	**34.64**†	**33.31**†
	THOR-MoE (Top-p)	27.63†	46.82†	24.15†	31.55†	29.97	44.18†	32.96†	34.26	32.91†

Table 2: Averaged BLEU scores on multilingual translation with encoder-decoder architectures. "†" denotes that statistically significant better than Lingual-MoE with t-test $p < 0.05$.

Models	IT	Koran	Medical	Law	Subtitles	Avg.
SFT-3B	86.39	73.17	85.43	87.19	78.52	82.14
THOR-MoE (Top-p)	87.31	74.09	86.09	87.88	79	82.87

Table 3: COMET scores on multi-domain translation benchmarks with decoder-only architecture.

	Multi-Domain	Multilingual
THOR-MoE (Top-p)	41.48	32.91
w/o hierarchical task-guided routing	40.16	31.57
w/o context-responsive routing	40.74	32.12

Table 4: Ablation Study with Avg. results.

shows the results where the 'Non-Mixed Representation' denotes that directly uses the automatically predict task label to extract corresponding task representation and 'Golden Representation' indicates using the golden task label to extract corresponding representation. We conclude that the task knowledge indeed has a positive impact on translation performance (vs. baseline). We also observe that using the automatically predicted task labels (actually the mixed task representation) shows better results than using ground truth and Non-Mixed manner in terms of Avg. BLEU scores. The reason may be that the mixed task representation has certain fault tolerance. We also analyze the accuracy of task prediction to further show its quality

	Multi-Domain	Multilingual
baseline (*w/o* any task Rep.)	40.74	32.12
Non-Mixed Rep.	40.95	32.44
Golden Rep.	41.32	32.78
THOR-MoE (Mixed manner)	41.48	32.91

Table 5: The Avg. results of different representation (Rep.) manner during hierarchical task-guided routing.

	Multi-Domain	Multilingual
baseline (*w/o* any task Rep.)	40.74	32.12
Infuse task Rep. into token Rep.	40.95	32.44
THOR-MoE (Hierarchical manner)	41.48	32.91

Table 6: The investigation of why hierarchical manner.

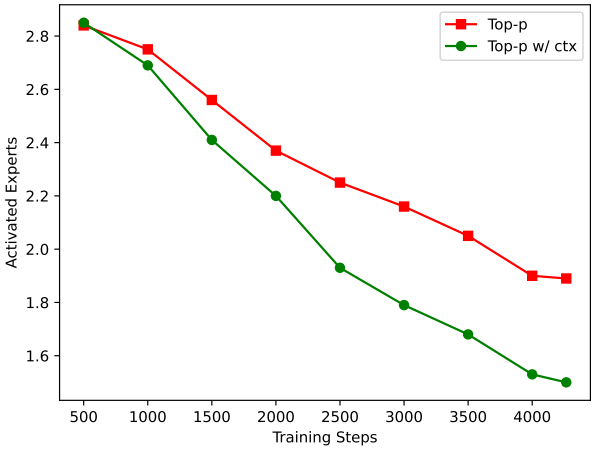

Figure 2: Average activated experts number across training steps on the multi-domain translation task.

in Appendix A.

6.3 Why Hierarchical Design?

In this section, we aim to investigate the manner of using mixed task representation. Table 6 shows that introducing task knowledge indeed helps tokens route well and achieve improvement (vs. baseline). The hierarchical manner greatly outperforms the direct infusion with token representation before routing, proving the effectiveness of hierarchical design, which unlock the potential of MoE with the help of the mixed task knowledge.

Domains	IT	Koran	Medical	Law	Subtitles	Avg.
Top-p	1.87	1.95	1.82	1.92	1.77	1.87
Top-p *w/* ctx	1.37	1.61	1.42	1.56	1.31	1.45

Table 7: Average activated experts (AE) in different domain translation tasks. 'Top-p *w/* ctx' denotes the proposed method to infuse the context during routing.

6.4 Analysis of Efficient Training and Inference

To further explore whether our proposed method is efficient in training and inference, we calculate the average number of experts activated by the model on multi-domain translation tasks. Figure 2 and Table 7 shows the average number of experts activated per token across various translation tasks during training and inference. The result is averaged across all the layers of transformers.

During training, we can see that our method converges faster to use less activated experts than the vanilla Top-p (Huang et al., 2024). During inference, we can observe that across all five tasks, the number of activated experts is less than original Top-p routing, averaging 1.45 activated experts (less than 22% activated parameters) with better performance. Both findings show that the context plays a key role in guiding token routing.

7 Related Work

Neural Machine Translation. The NMT have received remarkable attention in the era of LLMs (Jawahar et al., 2023b; Zhou et al., 2023). Previous work mainly focuses on the continual learning of new domains (Gu and Feng, 2020; Gu et al., 2022; Liang et al., 2024), introducing ready-made task-related (linguistics) knowledge via adapter (Zhang et al., 2020, 2021), parameter sharing (Aharoni et al., 2019), or task-specific modules (Pires et al., 2023), and multilingual representation learning (Yang et al., 2021).

Mixture-of-Experts. Jacobs et al. (1991) first proposes the concept of MoE, which consists of a series of network sub-modules. The Sparsely-gated MoE, as a variant, which only activates a few expert networks for each input, has been shown its superiority in diverse NLP and computer vision applications(Shazeer et al., 2017b; Zoph et al., 2022). In the context of NMT, most of previous work focuses on addressing the over/under-fitting (Elbayad et al., 2023) via regularization strategies (Elbayad et al., 2023) or modularizing MoE (Li et al., 2023c; Zhang et al., 2024), and incorporating task-related knowledge (language, domain *etc*) (Kudugunta et al., 2021; Zhao et al., 2024b; Pham et al., 2023; Gururangan et al., 2022; Li et al., 2023b), achieving impressive performance. Besides, there are some studies aim to improve the training/inference efficiency of routing via adaptive computation (Jawahar et al., 2023a) or reducing the number of activated experts (Li et al., 2023a; Huang et al., 2024).

Different from these prior works that directly use the task-related knowledge, we propose a hierarchical context-responsive routing method where we automatically extract corresponding domain/language knowledge and design a hierarchical network to guide the task-level routing. Besides, we aim to help each token accurately select specialized and suitable experts and thus we incorporate the context to guide the token routing rather than relying on the localized token only in existing work. In this manner, the context-responsive routing can improve the training or inference efficiency, which is compatible with previous Top-k and Top-p routing policies.

8 Conclusion

In this paper, we propose a new hierarchical task-gudied and context-responsive routing framework for NMT. To automatically obtain the task knowledge, we propose to predict it and then use mixed task representation. Consequently, we design a hierarchical routing at the task level and the token level. Further, we propose to inject context to enhance the effectiveness of token routing. Extensive experiments on multi-domain and multilingual translation benchmarks show the superiority and generalization of our proposed approach.

Limitations

While we introduce the mixed domain/language and context knowledge into routing in hierarchical manner and achieve good results, there are some limitations worth considering to study in future work: (1) In this study, the design relies on the prior (the number of tasks and language groups), which may limit its extention to broader topics (Li and Zhou, 2025); (2) This work only conduct experiments on translation tasks and does not conduct experiments other generation or discriminative tasks.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Mikko Aulamo and Jörg Tiedemann. 2019. The OPUS resource repository: An open package for creating parallel corpora and machine translation services. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics, NoDaLiDa 2019, Turku, Finland, September 30 - October 2, 2019*, pages 389–394.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Maha Elbayad, Anna Sun, and Shruti Bhosale. 2023. Fixing MoE over-fitting on low-resource languages in multilingual machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 14237–14253, Toronto, Canada. Association for Computational Linguistics.

Angela Fan, Shruti Bhosale, Holger Schwenk, Zhiyi Ma, Ahmed El-Kishky, Siddharth Goyal, Mandeep Baines, Onur Celebi, Guillaume Wenzek, Vishrav Chaudhary, Naman Goyal, Tom Birch, Vitaliy Liptchinsky, Sergey Edunov, Edouard Grave, Michael Auli, and Armand Joulin. 2020. Beyond english-centric multilingual machine translation. *CoRR*, abs/2010.11125.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Fabian Gloeckle, Badr Youbi Idrissi, Baptiste Rozière, David Lopez-Paz, and Gabriel Synnaeve. 2024. Better & faster large language models via multi-token prediction. *Preprint*, arXiv:2404.19737.

Shuhao Gu and Yang Feng. 2020. Investigating catastrophic forgetting during continual training for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4315–4326, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Shuhao Gu, Bojie Hu, and Yang Feng. 2022. Continual learning of neural machine translation within low forgetting risk regions. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1707–1718, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Suchin Gururangan, Mike Lewis, Ari Holtzman, Noah A. Smith, and Luke Zettlemoyer. 2022. DEMix layers: Disentangling domains for modular language modeling. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5557–5576, Seattle, United States. Association for Computational Linguistics.

Quzhe Huang, Zhenwei An, Nan Zhuang, Mingxu Tao, Chen Zhang, Yang Jin, Kun Xu, Kun Xu, Liwei Chen, Songfang Huang, and Yansong Feng. 2024. Harder task needs more experts: Dynamic routing in MoE models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12883–12895, Bangkok, Thailand. Association for Computational Linguistics.

Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. 1991. Adaptive mixtures of local experts. *Neural Comput.*, 3(1):79–87.

Ganesh Jawahar, Subhabrata Mukherjee, Xiaodong Liu, Young Jin Kim, Muhammad Abdul-Mageed, Laks Lakshmanan, V.S., Ahmed Hassan Awadallah, Sebastien Bubeck, and Jianfeng Gao. 2023a. AutoMoE: Heterogeneous mixture-of-experts with adaptive computation for efficient neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 9116–9132, Toronto, Canada. Association for Computational Linguistics.

Ganesh Jawahar, Subhabrata Mukherjee, Xiaodong Liu, Young Jin Kim, Muhammad Abdul-Mageed, Laks V. S. Lakshmanan, Ahmed Hassan Awadallah, Sébastien Bubeck, and Jianfeng Gao. 2023b. Automoe: Heterogeneous mixture-of-experts with adaptive computation for efficient neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 9116–9132. Association for Computational Linguistics.

Songtao Jiang, Tuo Zheng, Yan Zhang, Yeying Jin, Li Yuan, and Zuozhu Liu. 2024. Med-MoE: Mixture of domain-specific experts for lightweight medical vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 3843–3860, Miami, Florida, USA. Association for Computational Linguistics.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39.

Sneha Kudugunta, Yanping Huang, Ankur Bapna, Maxim Krikun, Dmitry Lepikhin, Minh-Thang Luong, and Orhan Firat. 2021. Beyond distillation: Task-level mixture-of-experts for efficient inference. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3577–3599, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021a. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen. 2021b. Gshard: Scaling giant models with conditional computation and automatic sharding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Jiamin Li, Qiang Su, Yitao Yang, Yimin Jiang, Cong Wang, and Hong Xu. 2023a. Adaptive gating in mixture-of-experts based language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3577–3587, Singapore. Association for Computational Linguistics.

Margaret Li, Suchin Gururangan, Tim Dettmers, Mike Lewis, Tim Althoff, Noah A. Smith, and Luke Zettlemoyer. 2023b. Branch-train-merge: Embarrassingly parallel training of expert language models.

Shangjie Li, Xiangpeng Wei, Shaolin Zhu, Jun Xie, Baosong Yang, and Deyi Xiong. 2023c. MMNMT: Modularizing multilingual neural machine translation with flexibly assembled MoE and dense blocks. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4978–4990, Singapore. Association for Computational Linguistics.

Ziyue Li and Tianyi Zhou. 2025. Your mixture-of-experts LLM is secretly an embedding model for free. In *The Thirteenth International Conference on Learning Representations*.

Yunlong Liang, Fandong Meng, Jiaan Wang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2024. Continual learning with semi-supervised contrastive distillation for incremental neural machine translation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10914–10928, Bangkok, Thailand. Association for Computational Linguistics.

Hai Pham, Young Jin Kim, Subhabrata Mukherjee, David P. Woodruff, Barnabas Poczos, and Hany Hassan. 2023. Task-based MoE for multitask multilingual machine translation. In *Proceedings of the 3rd Workshop on Multi-lingual Representation Learning (MRL)*, pages 164–172, Singapore. Association for Computational Linguistics.

Telmo Pires, Robin Schmidt, Yi-Hsiu Liao, and Stephan Peitz. 2023. Learning language-specific layers for multilingual machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14767–14783, Toronto, Canada. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of WMT*, pages 186–191.

Yiwei Qin, Xuefeng Li, Haoyang Zou, Yixiu Liu, Shijie Xia, Zhen Huang, Yixin Ye, Weizhe Yuan, Hector Liu, Yuanzhi Li, et al. 2024. O1 replication journey: A strategic progress report–part 1. *arXiv preprint arXiv:2410.18982*.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale. *Preprint*, arXiv:2201.05596.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017a. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc V. Le, Geoffrey E. Hinton, and Jeff Dean. 2017b. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net.

Qwen Team. 2024a. Qwen1.5-moe: Matching 7b model performance with 1/3 activated parameters".

Qwen Team. 2024b. Qwen2.5: A party of foundation models.

An Wang, Xingwu Sun, Ruobing Xie, Shuaipeng Li, Jiaqi Zhu, Zhen Yang, Pinxue Zhao, J. N. Han, Zhanhui Kang, Di Wang, Naoaki Okazaki, and Cheng zhong Xu. 2024. Hmoe: Heterogeneous mixture of experts for language modeling. *Preprint*, arXiv:2408.10681.

Junhong Wu, Yuchen Liu, and Chengqing Zong. 2024. F-MALLOC: Feed-forward memory allocation for continual learning in neural machine translation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7180–7192, Mexico City, Mexico. Association for Computational Linguistics.

Yilin Yang, Akiko Eriguchi, Alexandre Muzio, Prasad Tadepalli, Stefan Lee, and Hany Hassan. 2021. Improving multilingual translation by representation and gradient regularization. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7266–7279, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zihao Zeng, Yibo Miao, Hongcheng Gao, Hao Zhang, and Zhijie Deng. 2024. AdaMoE: Token-adaptive routing with null experts for mixture-of-experts language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 6223–6235, Miami, Florida, USA. Association for Computational Linguistics.

Biao Zhang, Ankur Bapna, Rico Sennrich, and Orhan Firat. 2021. Share or not? learning to schedule language-specific capacity for multilingual translation. In *International Conference on Learning Representations*.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020. Improving massively multilingual neural machine translation and zero-shot translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1628–1639, Online. Association for Computational Linguistics.

Fan Zhang, Mei Tu, Song Liu, and Jinyao Yan. 2024. A lightweight mixture-of-experts neural machine translation model with stage-wise training strategy. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 2381–2392, Mexico City, Mexico. Association for Computational Linguistics.

Hao Zhao, Zihan Qiu, Huijia Wu, Zili Wang, Zhaofeng He, and Jie Fu. 2024a. HyperMoE: Towards better mixture of experts via transferring among experts. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10605–10618, Bangkok, Thailand. Association for Computational Linguistics.

Xinyu Zhao, Xuxi Chen, Yu Cheng, and Tianlong Chen. 2024b. Sparse moe with language guided routing for multilingual machine translation. In *The Twelfth International Conference on Learning Representations*.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, and Zheyan Luo. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 400–410, Bangkok, Thailand. Association for Computational Linguistics.

Chulun Zhou, Yunlong Liang, Fandong Meng, Jinan Xu, Jinsong Su, and Jie Zhou. 2023. RC3: Regularized contrastive cross-lingual cross-modal pre-training. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 11747–11762, Toronto, Canada. Association for Computational Linguistics.

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. St-moe: Designing stable and transferable sparse expert models. *arXiv preprint arXiv:2202.08906*.

Simiao Zuo, Xiaodong Liu, Jian Jiao, Young Jin Kim, Hany Hassan, Ruofei Zhang, Jianfeng Gao, and Tuo Zhao. 2022. Taming sparsely activated transformer with stochastic experts. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

	Tasks/Groups	Train	Valid	Test
Multi-Domain Translation Dataset (De→En)	IT Koran Law Medical Subtitles	0.22M 18K 0.47M 0.25M 0.5M	2000	2000
Multilingual Translation Dataset	16	17,559,950	30*1000	30*1000

Table 8: The data statistic of the multi-domain translation dataset and multilingual translation (OPUS-16) dataset. The number in Train/Valid/Test columns denotes the number of sentence pairs in each domain/language pair.

Models	IT	Koran	Medical	Law	Subtitles	Avg.
Qwen1.5-MoE-A2.7B (SFT, Top-8)	45.59	23.31	55.67	60.18	29.21	42.79
THOR-MoE (Top-p, p=0.5)	46.76	24.03	55.9	61.13	30.29	43.62
Qwen1.5-MoE-A2.7B (SFT, Top-8)	87.3	74.24	85.85	87.78	79.36	82.91
THOR-MoE (Top-p, p=0.5)	87.45	74.58	86.45	88.75	80.23	83.49

Table 9: BLEU (top block) / COMET (below block) scores on multi-domain translation benchmarks with decoder-only architecture. Note that top-8 means it activates 4 shared experts and 4 non-shared experts out of 60 experts.

A Task Prediction

We also evaluate the performance of the *Task Predictor* to show whether the classifier can accurately predict suitable labels. The results are 82.45% and 64.89% for domain and language prediction, respectively. It suggests that our classifier can predict suitable labels and further provide effective mixed task representation for routing.

B Comparison to Larger Model

We implemented our proposed method based on a large-scale Qwen1.5-MoE-A2.7B (totally 14B, activated 2.7B, denoted as THOR-MoE). Besides, we also fine-tuned the vanilla Qwen1.5-MoE-A2.7B model as the baseline. The averaged COMET and BLEU results shown in Table 9 demonstrate that the THOR-MoE model at the 14B level also achieve better results than fine-tuned Qwen1.5-MoE-A2.7B model in terms of BLEU and COMET scores.

C Analysis of Domain Similarity

Following previous work (Wu et al., 2024), we conduct some analysis for domain similarity. Specifically, we sum the predicted soft label and then normalize it. The vertical axis and horizontal axis are unsupervised cluster label (clustering with BERT-base and k=5) and predicted soft labels, respectively. The similarity matrix are shown in Table 10.

	IT	Koran	Medical	Law	Subtitles
IT	0.833	0.027	0.055	0.0219	0.065
Koran	0.012	0.817	0.003	0.019	0.149
Medical	0.111	0.004	0.737	0.135	0.013
Law	0.142	0.008	0.032	0.798	0.02
Subtitles	0.058	0.076	0.009	0.006	0.851

Table 10: BLEU (top block) / COMET (below block) scores on multi-domain translation benchmarks with decoder-only architecture. Note top-8 means it activates 4 shared experts and 4 non-shared experts out of 60 experts.

The results show that there exist some sentences that are assigned to a cluster of another domain (i.e., hard cases). This makes sense as the mixed representation can capture such domain information, making the proposed approach works (the reason).