EIT: ENHANCED INTERACTIVE TRANSFORMER

Anonymous authors
Paper under double-blind review

ABSTRACT

In this work, we tackle the head degradation problem in attention. We propose an **E**nhanced **I**nteractive **T**ransformer (EIT) architecture in which the standard multi-head self-attention is replaced with the enhanced multi-head attention (EMHA). EMHA removes the one-to-one mapping constraint among queries and keys in multiple subspaces and allows each query to attend to multiple keys. On top of that, we develop a method to make full use of many-to-many mapping by introducing two interaction models, namely Inner-Subspace Interaction and Cross-Subspace Interaction. Extensive experiments on a wide range of tasks (e.g. machine translation, abstractive summarization, grammar correction, language modelling and brain disease automatic diagnosis) show its superiority with a very modest increase in model size.

1 INTRODUCTION

Transformer (Vaswani et al., 2017) and its variants have yielded promising results on a wide range of natural language processing tasks (Devlin et al., 2019; Brown et al., 2020). The strength lies in its ability to capture global dependencies among all positions of a given sequence, which is endowed by the multi-head self-attention network (MHSA). Just as its name implies, MHSA splits the hidden representation into several feature subspaces (namely head), and attention operations are performed within each subspace.

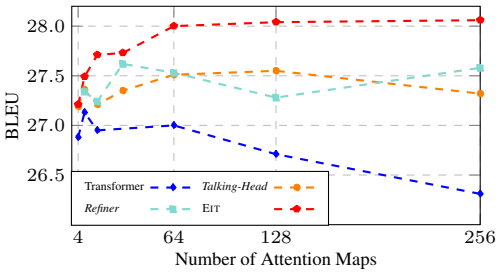


Figure 1: Performance of different models (pre-norm) across number of attention maps.

Researchers attribute the success of MHSA to the fact that different subspaces can capture distinct information, and further regularization of enlarging the head distance leads to good results (Li et al., 2018). Except for the empirical evidence, Wang et al. (2022) theoretically demonstrates that multi-head strategy can degrade the rate of over-smoothing[1] and the effect is in proportion to the number of subspaces. Since the benefits of MHSA, a natural question arises that ***"Can Transformer attain persistent and sufficient gains from larger subspaces?"***

For vanilla Transformer, simply enlarging subspaces suffers from performance deterioration since each query and key owns a lower dimension which can not guarantee precise attention maps (Shazeer et al., 2020). To overcome this downside, one can apply a tiny model, e.g., a linear layer, to the attention maps to learn more accurate attention maps. Along this research line, Zhou et al. (2021b) expands the attention distribution space based on original attention maps without reducing the head dimension, namely *Attention Expansion* mechanism. As is shown in Figure 1, we plot the performance against the number of heads on WMT'14 En-De. Obviously, the vanilla Transformer cannot make full use

Table 1: Comparison of different ways to obtain more heads. DIV, HD and SI denote diversity, head dimension and subspace interaction.

Method	**DIV**	**HD**	**SI**
Multi-Head Partition	Low	Low	Limited
Attention Expansion	Low	High	Limited
Many-to-Many	High	High	Sufficient

[1]Over-smoothing means the the similarity among representations of different tokens, which is directly related to the expressiveness of a model.

of the enlarging subspaces. Meanwhile, both *Talking-Head* and *Refiner* cannot persist the benefits when the head is larger than 32 or 128. Besides, their gains from enlarged subspaces are weak. This phenomenon could be interpreted from three aspects summarized in Table 1. First, in standard *Multi-Head Partition*, the head dimension descends as the number of heads increases, resulting in inaccurate attention maps. Second, the attention maps generated by *Attention Expansion* own high similarity, which deteriorates the quality of information contained in attention maps. These issues root the main cause for the difficulty of attaining more gains from enlarged subspaces.

In this work, we design a new variant, namely EIT, to solve the aforementioned weakness simultaneously. Concretely, we break the vanilla one-to-one mapping and instead adopt a *many-to-many* mapping strategy (Figure 2(b)) to acquire distinct subspace representation without sacrificing per head dimension. Moreover, we further develop a two-stage enhanced interactive approach to better utilize the enlarged subspaces. It involves two interaction models - call them *inner-subspace interaction* and *cross-subspace interaction*. The inner-subspace interaction model (ISI) encourages the learning of unique and refined features for each subspace, while the cross-subspace interaction model (CSI) focuses more on the learning of universal features over different subspaces.

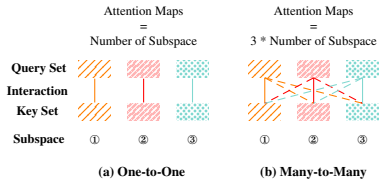


Figure 2: Attention models (3 heads) without and with interaction over heads. Many-to-Many can generate more diverse attention maps.

Our contributions can be summarized as follows:

- We propose a many-to-many mapping that generates numerous attention maps efficiently.
- We propose an EIT architecture that can outperform Transformer on a wide range of tasks including machine translation, abstractive summarization, grammar error correction, language modelling and brain disease automatic diagnosis.
- EIT can attain persistent gains from enlarged heads as well as keep high expressiveness. It can also simulate some property of attention with numerous heads, e.g., anti-oversmoothing. For example, features learned by EIT have a lower token correlation.
- The final attention maps generated by EIT are more confident in their decision though these attention maps own high similarity among each other.

2 ATTENTION IN TRANSFORMER

The core component of Transformer is the MHSA module. Given an embedded input sequence $\mathbf{X} \in \mathbb{R}^{T\times d}$, where T and d represent the sequence length and embedding dimension respectively. MHSA first generates three key components, namely queries, keys and values, via three linear transformation matrices $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ as: $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, $\mathbf{V} = \mathbf{X}\mathbf{W}_V$. It then divides the $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ into three sets U_Q, U_K and U_V, e.g., U_K denotes the set $\{\mathbf{K}^1, \cdots, \mathbf{K}^M\}$ and M is the number of heads. The attention distribution of i-th head is computed via: $\mathbf{A}^i = \mathrm{Softmax}(\frac{\mathbf{Q}^i(\mathbf{K}^i)^\top}{\sqrt{d_k}}) \in \mathbb{R}^{T\times T}$, where d_k is the head dimension that equals $\frac{d}{M}$. Then it calculates the final output $\mathbf{O}$ as follows:

$$\mathbf{O} = \left[\mathbf{O}^1, \ldots, \mathbf{O}^M\right] \mathbf{W}_O = \left[\mathbf{A}^1\mathbf{V}^1, \ldots, \mathbf{A}^M\mathbf{V}^M\right] \mathbf{W}_O \tag{1}$$

where $\mathbf{O}^i \in \mathbb{R}^{T\times d_k}$ is the output of i-th subspace in MHSA and $\mathbf{W}_O \in \mathbb{R}^{d\times d}$ is a learnable matrix.

Such a way to obtain attention maps has an important property: the head dimension is inversely proportional to the number of subspaces. Given this, when the number of subspaces is too large, the queries and keys with low head dimensions can not generate accurate attention maps (Shazeer et al., 2020). Consequently, Transformer suffers performance degradation. Previous work such as refiner (Zhou et al., 2021b) adopts *attention expansion* to generate more attention maps from a small number of original attention maps as follows: $\left[\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^{M^e}\right] = f(\left[\mathbf{A}^1, \mathbf{A}^2, \ldots, \mathbf{A}^M\right])$ where M^e is the number of heads after *attention expansion* and $f()$ is the expansion function which can be linear transformation or convolution. Note that the input attention maps are calculated via query-key multiplication. There is no guarantee to keep the diversity of the expanded attention maps, which will prevent the self-attention attaining more gains from enlarged subspaces (See Figure 6(a)).

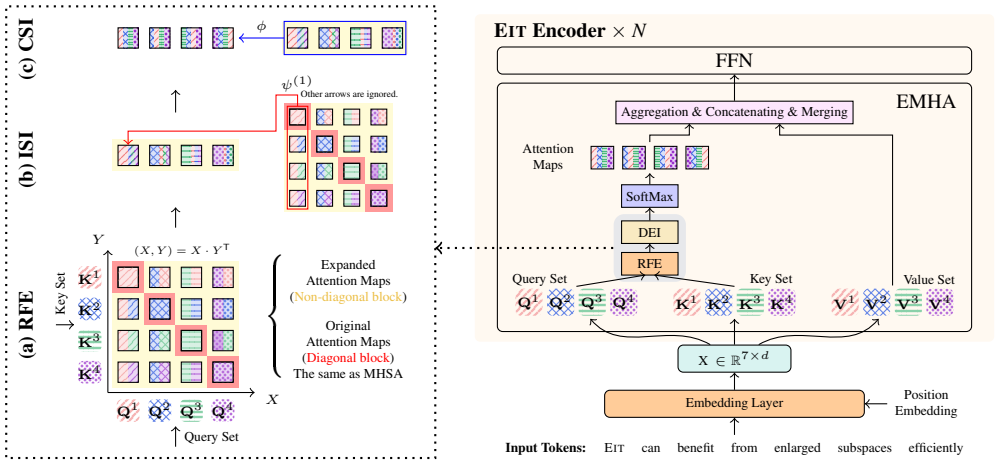


Figure 3: The model architecture of our proposed EIT (omit layernorm (Ba et al., 2016) and residual connection (He et al., 2016) for simplicity). Note that RFE denotes the Receptive Field Expansion module and DEI denotes the Dual Enhanced Interaction module. ISI denotes the Inner-Subspace Interaction and CSI denotes the Cross-Subspace Interaction. Here, we adopt the pre-normalization strategy considering the training stability under different configurations.

3 Enhanced Interactive Transformer

In this work, we propose a novel Enhanced Interactive Transformer in which we replace the multi-head self-attention with Enhanced Multi-Head Attention mechanism (EMHA) that can efficiently benefit from more heads. The proposed EMHA augments the MHSA from two aspects: 1) It adopts the many-to-may mapping schema to generate a large number of diverse attention maps. 2) It introduces comprehensive interactions to unearth the merits of the enlarged subspaces.

3.1 Definition of Interaction Space and Receptive Field

We first start with the definition of interaction space. Focusing on the attention operation, the interaction space, namely Φ, consists of several query-key mapping pairs, e.g., $\{\langle \mathbf{Q}^1, \mathbf{K}^1 \rangle, \ldots, \langle \mathbf{Q}^M, \mathbf{K}^M \rangle\}$. In this way, the size of the interaction space ($|\Phi|$) describes how many mapping pairs are there during the attention calculation. To investigate how to enlarge the interaction space effectively, another important concept, the receptive field of the attention mechanism, is defined. The receptive field refers to the set of keys each query can attend to. In MHSA, the receptive field of each query $\mathbf{Q}^i$ is $\{\mathbf{K}^i\}$. However, in EMHA, without special declaration, the receptive field of each query $\mathbf{Q}^i$ is the whole key set $\{\mathbf{K}^1, \ldots, \mathbf{K}^M\}$. The comparison of these two variants in terms of the receptive field and the interaction space is listed in Table 2. The connection between Φ and Ψ is formulated as: $|\Phi| = M \cdot |\Psi|$, where $|\Psi|$ denotes the size of receptive field. We also extend EMHA to a universal one with any size of receptive field for efficiency-performance trade-offs in Appendix H.

Table 2: Comparison of MHSA and EMHA in terms of two metrics. Here we set M to 8 as usual.

Module	$\lvert\Psi\rvert$	$\lvert\Phi\rvert$
MHSA	1	8
EMHA	8	64

3.2 Receptive Field Expansion

Through Section 3.1, we can easily connect the strength of interaction space with the number of attention maps since in MHSA, attention maps are always computed by applying $\mathrm{sim}(\cdot)$ to query-key mappings, e.g., dot-multiplication. Thus, it is crucial to efficiently enlarge the interaction space.

Many-to-Many Mapping Scheme To efficiently enlarge the interaction space, we propose to expand the receptive field of each query from the single key to M keys. As illustrated in Figure 3(a), four queries and four keys can be served as two components in a bipartite graph and each element, e.g., $\mathbf{Q}^1$, in a component can interact with any elements, e.g., $\mathbf{K}^1, \ldots, \mathbf{K}^4$, in another component. When the interaction operation is the scaled dot multiplication, we can obtain 16 attention maps,

which is four times as much as that in standard MHSA. Formally, supposing one with M heads, the i-th attention map can be formally calculated as:

$$\mathbf{S}^i = \frac{\mathbf{Q}^{\lfloor (i-1)/M+1 \rfloor} (\mathbf{K}^{(i-1)\%M+1})^\mathsf{T}}{\sqrt{d_k}} \qquad \forall i \in \{1, \ldots, M^2\} \tag{2}$$

where $\mathbf{S}^i \in \mathbb{R}^{T \times T}$ is the attention maps without softmax, $\lfloor \rfloor$ is the round down operation and $\%$ is the mod operation. For example, $\mathbf{S}^5$ is computed by $\mathbf{Q}^2$ and $\mathbf{K}^1$ when M equals to 4.

3.3 Dual Enhanced Interaction

The previous section has introduced how to efficiently enlarge the interaction space. However, directly enlarging the receptive field results in useless signals, e.g., noisy edges in attention maps. Besides, directly utilizing them decreases the dimension of value, degrading the expressiveness of features (Shazeer et al., 2020). A feasible solution is to distinguish the useful information and abandon the useless parts and fully leverage the useful ones.

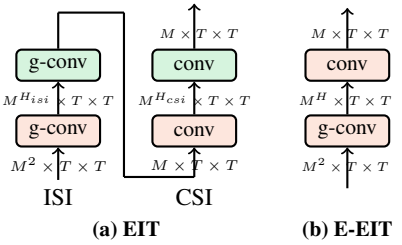


Figure 4: The architecture of ISI and CSI in our EIT and E-EIT. The ReLU is omitted.

3.3.1 Inner-Subspace Interaction

Definition of Inner-Subspace Interaction Relationship The inner-subspace interaction (ISI) relationship describes the connection among the attention maps generated by the same query. These attention maps have a closer relationship with each other since their queries are the same. This provides a suitable solution to distinguish the useful information and abandon the useless parts, since from the perspective of clustering, similar information can be grouped into one. As illustrated in Figure 3(a), the ISI relationship exists in the attention maps in the same column.

Inner-Subspace Interaction Modeling In ISI modeling, one thing we need to consider is to keep the diversity among the output attention maps when extracting useful information from these materials obtained by RFE. Given this, we adopt two-layer group convolutions (Krizhevsky et al., 2012) accompanied by the ReLU activation to consist this sub-module, since group convolution owns several separate convolutions to process features from different groups and subspace-specific functions (Cui et al., 2019) are of great benefit to enlarge head distances in the MHSA.

Denote $\mathbf{f}(\cdot)$ as a single layer group convolution. As illustrated in Figure 4(a), given the output of RFE, namely $\mathbf{S}$, as input, ISI sub-module is computed as:

$$\hat{\dot{\mathbf{S}}} = \mathbf{ReLU}(\mathbf{f}^{(\mathbf{0})}(\mathbf{S})), \qquad \dot{\mathbf{S}} = \mathbf{f}^{(\mathbf{1})}\left(\hat{\dot{\mathbf{S}}}\right), \tag{3}$$

where $\hat{\dot{\mathbf{S}}} \in \mathbb{R}^{M^{H_{isi}} \times T \times T}$ is the hidden output and $\dot{\mathbf{S}} \in \mathbb{R}^{M \times T \times T}$ is the final output of the ISI sub-module. We use $M^{H_{isi}}$ to represent the intermediate head size in ISI sub-module and set the number of groups in group convolutions to M. For other configurations, please see Appendix A.

3.3.2 Cross-Subspace Interaction

Definition of Cross-Subspace Interaction Relationship The cross-subspace interaction (CSI) relationship describes the collaboration among different subspaces, which aims to fully leverage the useful information. It exists in the attention maps generated by different queries. As illustrated in Figure 3(a), the CSI relationship exists in the attention maps in the different columns.

Cross-Subspace Interaction Modeling In CSI modeling, we want to leverage the advantages of attention maps of different heads to generate final precise attention maps. we adopt two-layer convolutions accompanied by the ReLU activation to consist this sub-module.

Let us denote $\mathbf{g}(\cdot)$ as a single layer convolution. As illustrated in Figure 4(a), given the output of ISI, namely $\dot{\mathbf{S}}$, as input, CSI sub-module is computed as:

$$\hat{\ddot{\mathbf{S}}} = \mathbf{ReLU}(\mathbf{g}^{(\mathbf{0})}\left(\dot{\mathbf{S}}\right)), \qquad \ddot{\mathbf{S}} = \mathbf{g}^{(\mathbf{1})}\left(\hat{\ddot{\mathbf{S}}}\right), \tag{4}$$

where $\hat{\ddot{\mathbf{S}}} \in \mathbb{R}^{M^{H_{csi}} \times T \times T}$ is the hidden output and $\ddot{\mathbf{S}} \in \mathbb{R}^{M \times T \times T}$ is the final output of the CSI sub-module. We use $M^{H_{csi}}$ to represent the intermediate head size in CSI sub-module.

3.4 EFFICIENT VERSION OF EIT

Despite the theoretically computational efficiency and parametric efficiency of group convolutions, they slow down the training in practice (Ma et al., 2018). To alleviate this issue, we provide another efficient version of EIT, namely E-EIT, by simplifying the design of ISI and CSI. In this way, E-EIT avoids parts of memory consumption and somehow speeds up the computational efficiency. Specifically, we adopt a single layer in each sub-module, as illustrated in Figure 4(b). In E-EIT, the dual enhanced interactions are computed as:

$$\dot{\mathbf{S}} = \mathbf{ReLU}(\mathbf{f}^{(\mathbf{0})}(\mathbf{S})), \qquad \ddot{\mathbf{S}} = \mathbf{g}^{(\mathbf{0})}\left(\dot{\mathbf{S}}\right), \tag{5}$$

where $\dot{\mathbf{S}} \in \mathbb{R}^{M^H \times T \times T}$ and $\ddot{\mathbf{S}} \in \mathbb{R}^{M \times T \times T}$ and M^H is a hyper-parameter, e.g. we set it as 32 for the base configuration.

4 EXPERIMENTS

4.1 RESULTS ON MACHINE TRANSLATION

We evaluated EIT on both WMT'14 En-De and WMT'16 En-Ro machine translation tasks with BLEU and sacreBLEU [2]. More details about the experimental setup are included in Appendix A.

Main Results Table 3 and Table 4 display the results of baselines and our methods on En-De and En-Ro tasks. For the En-De task, we select baselines from three aspects: head modification (Zhou et al., 2021b; Shazeer et al., 2020; Wang & Tu, 2020; Li et al., 2019; 2018; Zhang et al., 2022; Nguyen et al., 2022), localness modelling (Li et al., 2022; Fan et al., 2021; Yang et al., 2019), and deep transformer (Liu et al., 2020b; Wang et al., 2019; Wei et al., 2020). A similar selection of baselines is also for the En-Ro task. For a fair comparison, we re-implement some of these models in our codebase. We keep the training setups of these methods the same as ours. Through the broad comparison, we can get the following observations:

1. Our EIT variants outperform the vanilla Transformer with negligible added parameters against different configurations on both datasets. To be honest, EIT adopts frequent feed-forward-like mapping strategies, which is time consuming. To alleviate this, we also provide another choice, namely E-EIT. Beyond our expectations, it not only delivers competitive results compared with the full version, but also saves redundant computations (we show the comparison in Section 5.4). Besides, we also provides a lot of parameter analysis (Appendix H) and trade-off strategy (Appendix I).

2. Our EIT can beat all selected methods of head modification and localness modelling, including the latest methods such as MoA (Zhang et al., 2022), Fishformer (Nguyen et al., 2022), UMST (Li et al., 2022), on both datasets. This indicates that our methods can better unearth the potential of the multi-head strategy.

3. Our EIT can beat all selected deep transformers on the En-De task. For example, a 48-layer EIT can beat the 48-layer baseline over 0.69 BLEU points. Besides, compared to MSC and ADMIN, EIT beats them with fewer parameters.

4.2 RESULTS ON OTHER SEQUENCE GENERATION TASKS

We also conducted experiments on the other two sequence generation tasks: the abstractive summarization task and the grammar error correction task. Due to the limited page, please refer to the Appendix A for the details of experimental setups.

Table 5 shows that both EIT and E-EIT can outperform the standard Transformer by a large margin on both tasks. For example, EIT achieves scores of 41.62 ROUGE-1 points, 18.70 ROUGE-2 points and 38.33 ROUGE-L points on the CNN-DailyMail dataset and scores of 57.05 $\mathbf{F}_{0.5}$ on CONLL

[2] BLEU+case.mixed+numrefs.1+smooth.exp+ tok.13a+version.2.0.0

Table 3: BLEU and sacreBLEU points on WMT En-De Task. More details about the settings of thees models are given in Appendix.

Type	Model	WMT'14 En-De		
		θ (M)	BLEU	sBLEU
Head Modification	Refiner (2021b)	-	27.62	-
	Talking-Head (2020)	-	27.51	-
	Collaboration (2020)	-	27.55	-
	DyRouting (2019)	297M	28.96	-
	Disagree (2018)	-	29.28	-
	MoA (2022)	200M	29.40	-
	FISHformer (2022)	-	29.57	-
Localness	Dman (2021)	211M	28.97	27.8
	Csan (2019)	-	28.74	-
	Umst (2022)	242M	29.75	-
Deep Transformer	DLCL (2019)	-	29.30	-
	Admin (2020b)	262M	30.10	-
	Msc (2020)	272M	30.19	-
Our System (Pre-Norm)	Transformer base	62M	27.13	26.0
	EIT base	62M	**28.00**	26.9
	E-EIT base	62M	27.72	26.7
	Transformer deep-48	194M	29.60	28.5
	EIT deep-48	194M	**30.25**	29.2
	E-EIT deep-48	194M	30.16	29.1
	Transformer big	211M	28.80	27.7
	EIT big	212M	**29.79**	28.7
	E-EIT big	211M	29.61	28.5

Table 4: BLEU points on WMT En-Ro Task. More details about the settings of thees models are given in Appendix.

Type	Model	WMT'16 En-Ro	
		θ (M)	BLEU
Basic Baseline	ConvS2S (2017)	-	29.90
	Transformer (2020c)	-	34.30
	Transformer (2020)		34.16
	FlowSeq(2019)	-	31.97
	Int-TF (2021)	-	32.60
	DELIGHT (2021)	53M	34.70
Head modification	Refiner (2021b)	54M	34.25
	Talking-Head (2020)	54M	34.35
	Collaboration (2020)	54M	34.64
	FISHformer (2022)	49M	34.42
	MoA (Zhang et al., 2022)	56M	34.39
Localness	Dman (Fan et al., 2021)	-	34.49
	Umst (Li et al., 2022)	60M	34.81
Our System (Pre-Norm)	Transformer base	54M	34.23
	EIT base	54M	**35.10**
	E-EIT base	54M	35.01
	Transformer deep-24	111M	35.00
	EIT deep-24	111M	**35.40**
	E-EIT deep-24	111M	35.35
	Transformer big	196M	34.44
	EIT big	196M	**34.91**
	E-EIT big	196M	34.67

Table 5: Results on the summarization task and the correction task.

Model	CNN-DailyMail			CONLL		
	RG-1	RG-2	RG-L	Precision	Recall	$\mathbf{F}_{0.5}$
Transformer ‡	40.84	18.00	37.58	64.84	**36.61**	56.18
PG-Net (See et al., 2017)	39.53	17.28	36.38	-	-	-
MADY (Wang et al., 2021)	40.72	17.90	37.21	-	-	-
Dman (Fan et al., 2021)	40.98	18.29	37.88	-	-	-
BOTTOM-UP (Gehrmann et al., 2018)	41.22	18.68	**38.34**	-	-	-
SURFACE (Liu et al., 2021)	41.00	18.30	37.90	66.80	35.00	56.60
EIT	**41.62**	**18.70**	38.33	**69.98**	32.80	57.05
E-EIT	41.58	18.63	38.28	69.85	33.36	**57.31**

Table 6: AUROC, ACC, SEN and SPE points on ABIDE task.

Model	AUROC	ACC	SEN	SPE
MvS-GCN (Wen et al., 2022)	69.0	69.4	69.3	64.5
BrainNetTF (Kan et al., 2022)	80.9±2.6	71.8±3.0	71.1±4.1	72.5±1.9
BrainNetEITF	**81.3±2.7**	**73.8±3.2**	**73.9 ± 5.8**	**75.6 ± 4.7**

Table 7: Comparison of test Perplexity on wiki103-Text.

Model	testPPL
Transformer (Baevski & Auli, 2019)	22.50
EIT	**21.34**

dataset. Compared with other strong baselines, our EIT and E-EIT can still show superiority on these datasets in terms of ROUGE-1 points and $\mathbf{F}_{0.5}$ value, e.g., EIT surpasses SURFACE by 0.62 and 0.45 in terms of ROUGE-1 points and $\mathbf{F}_{0.5}$ value, respectively.

4.3 Results on Automatic Disease Diagnosis Task

We also evaluate our EIT on an automatic disease diagnosis task with four metrics: Accuracy, AUROC, Sensitivity and Specificity. We select a widely used real-world fMRI dataset: Autism Brain Imaging Data Exchange (ABIDE). ABIDE consists of 1009 brain networks from 17 international sites, of which 516 samples are autism spectrum disorder patients. Due to the strong heterogeneity of this data, we follow the preprocessing process in Kan et al. (2022). Moreover, we adopt the CC200 (Craddock et al., 2012) as the Regin-of-Interest (ROI) partition template. We select two latest methods, the Mvs-GCN (Wen et al., 2022) and BrainNetTF (Kan et al., 2022), as our comparison. The experiment setups and configurations of our BrainNetEITF are the same as in Kan et al. (2022). Each experiment is conducted 5 times and we report the mean and standard deviation of the four metrics. The results are exhibited in Table 6. We can see that our BrainNetEITF can outperform all the baselines in terms of all metrics.

4.4 RESULTS ON LANGUAGE MODELING

We also evaluate our EIT on a language modeling task: wiki103-text with Perplexity (PPL). We follow the official preprocessing procedure (Ott et al., 2019), and select Adaptive Input Transformer (Baevski & Auli, 2019) as our baselines. Both the baseline and our EIT are all 8-layer big models with 8 heads. More details are given in Appendix A. Table 7 displays the results. We can see that our EIT can also outperform the baseline with 1.16 in terms of testPPL.

5 DISCUSSION

In this section, we present a detailed analysis to provide some insights into why EMHA is superior to vanilla MHSA. Without the special declaration, we conduct experiments on EIT.

5.1 ABLATION STUDIES

Table 8 summarizes the impacts of removing each module on En-De and En-Ro tasks, respectively. Notably, when removing the RFE module (#2 vs. #3), we observe an obvious decline in performance on two translation tasks. This indicates that the RFE module plays an important role in EIT, as it can generate plentiful diverse attention maps. Besides, we can see there is a vast drop in BLEU when removing the ISI submodule (#2 vs. #4). This is because directly enlarging subspaces leads to the existence of some useless information. However, the ISI sub-module provides a suitable way to abandon it while retaining the benefits of the former. This unique design differs from the head expansion (Zhou et al., 2021b).Moreover, applying a single module (# 6, 7, 8 v.s #1, 2) is inferior to EIT, which indicates EIT is well-designed, e.g., ISI efficiently utilizes the enlarged heads and CSI fully learns universal features across all subspaces. Note that the additional primary cost comes from the ISI sub-module due to the unfriendly support for the implementation of group convolution in PyTorch (Paszke et al., 2019).

Table 8: Ablation study on two tasks. Time denotes the training computing time.

#	RFE	ISI	CSI	En-De		En-Ro	
				BLEU	Time	BLEU	Time
1				27.13	-	34.23	-
2	✓	✓	✓	**28.00**	1.45×	**35.10**	1.38×
3		✓	✓	27.39	1.15×	34.71	1.10×
4	✓		✓	25.79	1.22×	32.50	1.21×
5	✓	✓		27.70	1.40×	34.53	1.29×
6	✓			26.01	1.06×	30.67	1.05×
7		✓		27.32	1.30×	34.53	1.25×
8			✓	27.29	1.10×	34.55	1.06×

5.2 DOES IMPROVEMENTS COME FROM THE HEAD MODIFICATION

In our dual enhanced interaction modeling, we apply convolution operation to attention maps, which has a potential to introduce local biases, a kind of useful information for Transformer. To validate this, inspired by Fan et al. (2021), we define a localness metric, namely $\mathcal{C}$, that measures the localness of attention maps since if there is a local bias, each token will distribute larger attention weights on their neighbouring tokens.

We plot the $\mathcal{C}$ value of our EIT and Transformer on the test set of En-De task and CNN-DailyMail task in Figure 5. Note that due to the limit time of rebuttal, we only use a subset of test set consists of 1000 samples for CNN-DailyMail task. Through the results (mean), we can see that there is no significant local enhancement phenomenon on both tasks. Note that the attention maps in the first layer of EIT on the abstractive summarization have a strong local pattern, but the kernel sizes are all set to 1 on this task. So we conclude the improvements do not come from explicitly introducing local biases.

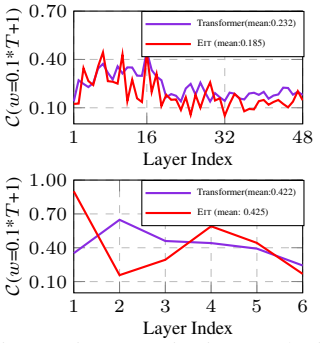


Figure 5: Quantitative analysis on localness in attention maps on En-De task (Above) and CNN-DailyMail task (Bottom).

5.3 WHAT CAN EIT BRING

In this section, we aim to unearth *why* EIT *is superior to standard transformers.* To be specific, we trace the variation of head distance among different heads throughout the calculation process, and

investigate the connection between head distance and quality of features. The results are exhibited in the Figure 6. For more details, please refer to Appendix B. We obtain the following observations:

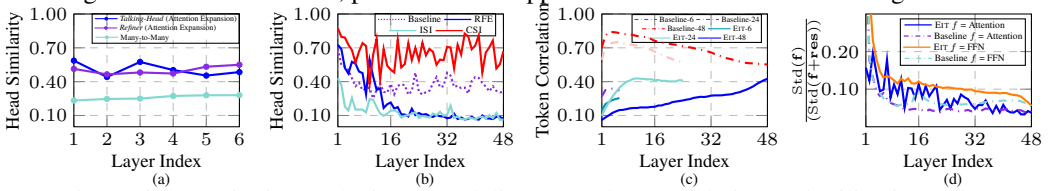


Figure 6: Quantitative analysis on head distance, token correlation and utilization ratio.

Diverse Initial Attention Maps As plotted in Figure 6(a), the initial numerous attention maps in our EIT (generated by RFE) own the lowest similarity, compared to that in Talking-head and Refiner (generated by attention expansion). This is within our expectation. When we further replace RFE with attention expansion in our EIT system, the performance on En-De task suffers a vast drop from 28.00 to 27.41. This indicates when one benefits from enlarged subspaces, he first needs to generate diverse initial numerous attention maps. They are just like raw materials, the level of diversity among them determines how much information they contain.

High-Similarity Final Attention Maps As plotted in Figure 6(b), EIT owns the higher average similarity among attention maps from different heads, compared to standard Transformer. Different from normal redundant attention maps, the high similarity is the result of head interaction and the high-similarity attention maps have a property: they are more confident in their decisions (see Section 5.5 and Appendix L). When we explicitly break this kind of high similarity via Li et al. (2018), the performance suffers from degradation (from 28.00 to 27.65 under base configuration). Thus, we say that the high similarity among attention maps resulting from head interaction is a good pattern.

Lower Token Correlation Figure 6(c) plots the token correlation of our EIT and Transformer models against encoder depths. The token correlation is measured by the pearson correlation coefficient (Benesty et al., 2009). More details are given in Appendix B. We can see that the features from EIT have a lower token correlation than that of vanilla Transformer. This indicates EIT has a more powerful learning capacity since high correlation among token representations reduces the learning capacity of Transformer (Gong et al., 2021; Shi et al., 2022; Wang et al., 2022).

More Efficient utilization of Functions Figure 6(d) plots the ratio of function output and final output of our EIT and Transformer models. We can see that the EMHA and FFN in EIT contribute more. This is unusual since Attention and FFN have a tendency to learn similar token representations and the residual connection is of great benefit to resist this tendency (Wang et al., 2022). A reasonable explanation is that EMHA and FFN in EIT has a tendency to learn lower similar token representation. So they do not rely too much on the residual connection.

5.4 Model Scalability and Efficiency Comparison

EIT is well scalable to deep configurations. The results are exhibited in Figure 7(left). EIT outperforms the Transformer at all depths with up to 0.69 BLEU points on average. Similar phenomena could be observed in E-EIT. Moreover, Figure 7(middle and right) also displays the memory consumption and computational cost. EIT costs 8.5% more memory consumption and 44.4% more training costs than the baseline with a depth of 6. The E-EIT only costs 9.4% more memory consumption and 21.7% more training costs than the baseline under all the configurations on average. This demonstrates E-EIT is efficient. Moreover, as aforementioned, the main extra costs come from the group convolution. We ascribe it to the unfriendly support by PyTorch. More analyses for this are included in the Appendix E. We will release an efficient implementation in future.

5.5 Visualization

We visualize the attention maps (Figure 8) for an example on the En-De task, and get the following observations: 1) Attention maps obtained by EIT attend to certain patterns, e.g., word boundary information (a@@, gh@@ and ast) (Li et al., 2022). 2) Our attention maps are more confident in their decision, e.g., each token pays more attention to a few positions rather than learning a smoothing view over all positions. These demonstrate that our EIT can generate more precise attention maps. Besides, though the nearly consistent attending results lead to the high similarity among attention

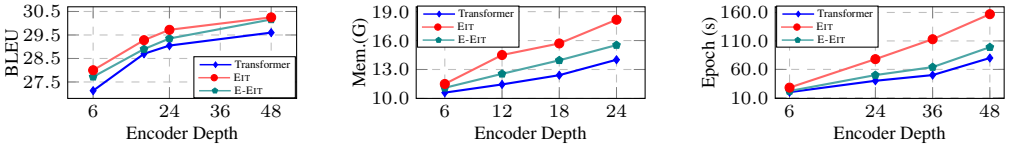


Figure 7: BLEU, memory and speed vs. encoder depth. Both EIT and E-EIT beat Transformer under all configurations. E-EIT can achieve comparable results with fewer training costs than EIT.

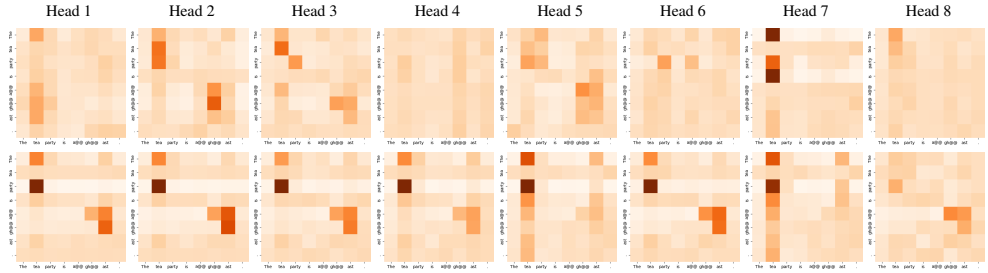


Figure 8: Quantitative examples of the attention distribution over a real case in WMT'14 En-De task. The above is the distribution generated by the standard Transformer, and the below is ours (EIT). Dark color means a higher value in the distribution.

maps, we find that if the patterns are useful, high-similarity attention maps are not a drawback. More results of other cases in other tasks are included in Appendix L.

6 Related Work

Improved Multi-Head Mechanism How to effectively improve the multi-head attention has always been a hot research topic in recent years. Previous work has revealed that multi-head attention can be further enhanced by encouraging individual attention heads to extract distinct information (Li et al., 2018; Cui et al., 2019; Sukhbaatar et al., 2019; Guo et al., 2020; Hao et al., 2019). Another branch of research is designing more complex interactive modeling to make better use of the multiple subspace information (Shazeer et al., 2020; Wang & Tu, 2020; Li et al., 2019). Besides, Voita et al. (2019) empirically demonstrates that some heads in attention are useless and can be pruned without performance degradation. Along this line, researchers investigate how to efficiently cut off redundant heads (Michel et al., 2019; Behnke & Heafield, 2020). Different from these work, Wang et al. (2022) theoretically demonstrates the benefits of multi-head attention. Motivated by it, our work investigates how to benefit from enlarged heads efficiently.

Convolution + Attention Convolution and attention mechanism have been dominated paradigms for local modelling and global modelling, respectively. Collaboration of convolution and attention mechanism has become an interesting topic. Recently, many researchers have shifted their attention to incorporating the convolution into the transformer (Yang et al., 2019; Zhou et al., 2021b; Zhao et al., 2019; Pan et al., 2021; Wu et al., 2021; Xiao et al., 2021; Peng et al., 2021). Their core idea is to fully leverages the advantages of these two paradigms. Our work follows this thread of research but is quite different from them in two aspects: 1) We mainly adopt the convolutions and group convolutions for flexibly modelling the cross-head interaction but not for injecting localness. 2) The convolution of our work is directly applied to the attention map. Note that Zhou et al. (2021b) is similar to our work. However, they mainly focus on modelling the localness bias.

7 Conclusions

In this paper, we propose EIT, an alternative to the standard multi-head attention network. It further advances the multi-head schema by breaking the standard one-to-one mapping constraint. Meanwhile, EIT employs the inner-subspace interaction and cross-subspace interaction to make full use of the expanded attention maps. In addition, E-EIT can be served as another choice considering the trade-off between performance and computation efficiency. Experimental results on five widely-used tasks demonstrate the effectiveness of EIT-variants, which deliver consistent improvements to the standard Transformer.

REFERENCES

Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, 2016.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations*, 2019.

Maximiliana Behnke and Kenneth Heafield. Losing heads in the lottery: Pruning transformer attention in neural machine translation. In *Proc. of EMNLP*, 2020.

Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. *Pearson Correlation Coefficient*. Springer Berlin Heidelberg, 2009.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In *Proc. of NeurIPS*, 2020.

Shamil Chollampatt and Hwee Tou Ng. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Proc. of AAAI*, 2018.

R Cameron Craddock, G Andrew James, Paul E Holtzheimer III, Xiaoping P Hu, and Helen S Mayberg. A whole brain fmri atlas generated via spatially constrained spectral clustering. *Human brain mapping*, 2012.

Hongyi Cui, Shohei Iida, Po-Hsuan Hung, Takehito Utsuro, and Masaaki Nagata. Mixed multi-head self-attention for neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, 2019.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*, 2019.

Zhihao Fan, Yeyun Gong, Dayiheng Liu, Zhongyu Wei, Siyuan Wang, Jian Jiao, Nan Duan, Ruofei Zhang, and Xuanjing Huang. Mask attention networks: Rethinking and strengthen transformer. In *Proc. of NAACL*, 2021.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International conference on machine learning*, 2017.

Sebastian Gehrmann, Yuntian Deng, and Alexander Rush. Bottom-up abstractive summarization. In *Proc. of EMNLP*, 2018.

Chengyue Gong, Dilin Wang, Meng Li, Vikas Chandra, and Qiang Liu. Vision transformers with patch diversification. 2021.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. Multi-scale self-attention for text classification. In *Proc. of AAAI*, 2020.

Jie Hao, Xing Wang, Shuming Shi, Jinfeng Zhang, and Zhaopeng Tu. Multi-granularity self-attention for neural machine translation. In *Proc. of EMNLP*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of CVPR*, 2016.

Xuan Kan, Wei Dai, Hejie Cui, Zilong Zhang, Ying Guo, and Carl Yang. Brain network transformer. In *Advances in Neural Information Processing Systems*, 2022.

Jungo Kasai, James Cross, Marjan Ghazvininejad, and Jiatao Gu. Non-autoregressive machine translation with disentangled context transformer. In *International conference on machine learning*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. of NeurIPS*, 2012.

Bei Li, Tong Zheng, Yi Jing, Chengbo Jiao, Tong Xiao, and Jingbo Zhu. Learning multiscale transformer models for sequence generation. In *Proc. of ICML*, 2022.

Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. Multi-head attention with disagreement regularization. In *Proc. of EMNLP*, 2018.

Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R. Lyu, and Zhaopeng Tu. Information aggregation for multi-head attention with routing-by-agreement. In *Proc. of NAACL*, 2019.

Ye Lin, Yanyang Li, Tengbo Liu, Tong Xiao, Tongran Liu, and Jingbo Zhu. Towards fully 8-bit integer inference for the transformer model. In *Proc. of IJCAI*, 2021.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proc. of EMNLP*, 2020a.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. Very deep transformers for neural machine translation. *CoRR*, 2020b.

Xuebo Liu, Longyue Wang, Derek F. Wong, Liang Ding, Lidia S. Chao, and Zhaopeng Tu. Understanding and improving encoder layer fusion in sequence-to-sequence learning. In *Proc. of ICLR*, 2021.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 2020c.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In *Proc. of ECCV*, 2018.

Xuezhe Ma, Chunting Zhou, Xian Li, Graham Neubig, and Eduard Hovy. FlowSeq: Non-autoregressive conditional sequence generation with generative flow. 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.

Sachin Mehta, Marjan Ghazvininejad, Srinivasan Iyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. Delight: Deep and light-weight transformer. In *Proc. of ICLR*, 2021.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? In *Proc. of NeurIPS*, 2019.

Tan Minh Nguyen, Tam Minh Nguyen, Hai Ngoc Do, Khai Nguyen, Vishwanath Saragadam, Minh Pham, Nguyen Duy Khuong, Nhat Ho, and Stanley Osher. Improving transformer with an admixture of attention heads. In *Advances in Neural Information Processing Systems*, 2022.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proc. of NAACL*, 2019.

Xuran Pan, Chunjiang Ge, Rui Lu, Shiji Song, Guanfu Chen, Zeyi Huang, and Gao Huang. On the integration of self-attention and convolution. *CoRR*, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of NeurIPS*. 2019.

Zhiliang Peng, Wei Huang, Shanzhi Gu, Lingxi Xie, Yaowei Wang, Jianbin Jiao, and Qixiang Ye. Conformer: Local features coupling global representations for visual recognition. In *Proc. of ICCV*, 2021.

Abigail See, Peter J. Liu, and Christopher D. Manning. Get to the point: Summarization with pointer-generator networks. In *Proc. of ACL*, 2017.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*, 2016.

Noam Shazeer, Zhenzhong Lan, Youlong Cheng, Nan Ding, and Le Hou. Talking-heads attention. *CoRR*, 2020.

Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen M. S. Lee, and James T. Kwok. Revisiting over-smoothing in BERT from the perspective of graph. In *Proc. of ICLR*, 2022.

Sainbayar Sukhbaatar, Edouard Grave, Piotr Bojanowski, and Armand Joulin. Adaptive attention span in transformers. In *Proc. of ACL*, 2019.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of NeurIPS*, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. In *Proc. of ACL*, 2019.

Huadong Wang and Mei Tu. Enhancing attention models via multi-head collaboration. In *International Conference on Asian Language Processing, IALP 2020, Kuala Lumpur, Malaysia, December 4-6, 2020*, 2020.

Lihan Wang, Min Yang, Chengming Li, Ying Shen, and Ruifeng Xu. Abstractive text summarization with hierarchical multi-scale abstraction modeling and dynamic memory. In *Proc. of SIGIR*, 2021.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *Proc. of ICLR*, 2022.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *Proc. of ACL*, 2019.

Xiangpeng Wei, Heng Yu, Yue Hu, Yue Zhang, Rongxiang Weng, and Weihua Luo. Multiscale collaborative deep models for neural machine translation. In *Proc. of ACL*, 2020.

Guangqi Wen, Peng Cao, Huiwen Bao, Wenju Yang, Tong Zheng, and Osmar Zaiane. Mvs-gcn: A prior brain structure learning-guided multi-view graph convolution network for autism spectrum disorder diagnosis. *Computers in Biology and Medicine*, 2022.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proc. of ICCV*, 2021.

Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross B. Girshick. Early convolutions help transformers see better. *CoRR*, 2021.

Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. Convolutional self-attention networks. In *Proc. of NAACL*, 2019.

Xiaofeng Zhang, Yikang Shen, Zeyu Huang, Jie Zhou, Wenge Rong, and Zhang Xiong. Mixture of attention heads: Selecting attention heads per token. *CoRR*, 2022.

Guangxiang Zhao, Xu Sun, Jingjing Xu, Zhiyuan Zhang, and Liangchen Luo. MUSE: parallel multi-scale attention for sequence to sequence learning. *CoRR*, 2019.

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *CoRR*, 2021a.

Daquan Zhou, Yujun Shi, Bingyi Kang, Weihao Yu, Zihang Jiang, Yuan Li, Xiaojie Jin, Qibin Hou, and Jiashi Feng. Refiner: Refining self-attention for vision transformers. *CoRR*, 2021b.

A SETUP OF EXPERIMENTS

A.1 MACHINE TRANSLATION TASK

Dataset We evaluated our approach on two widely used machine translation datasets: WMT'14 En-De and WMT'16 En-Ro. The En-De dataset contains approximately 4.5M tokenized training sentence pairs. We selected newstest2013 and newstest2014 as the validation and test data, respectively. As for the En-Ro dataset, it consists of 0.6M tokenized training sentence pairs. We performed shared BPE operations on both datasets to overcome the out-of-vocabulary (OOV) problem. Concretely, we set the size of BPE operations to 32K and 20K for En-De and En-Ro datasets, resulting in a shared vocabulary with sizes of 34040 and 19064, respectively.

Model Configuration Our model architectures are based on Transformer (Vaswani et al., 2017). We provided three basic configurations, namely *base*, *deep*, and *big* which follow the configurations in Vaswani et al. (2017). We adopted a pre-normalization strategy (Wang et al., 2019) considering training stability under different configurations. The newly added hyper-parameters follow the settings in Table 10.

Training & Evaluation Our implementations are based on Fairseq (Ott et al., 2019). Our experiments are performed on the GEFORCE RTX 3090 cards. We use 8 GEFORCE RTX 3090 cards to train models for the WMT'14 En-De task. As for the models on the WMT'16 En-Ro task, we train them on 4 GEFORCE RTX 3090 cards. The batch sizes for En-De and En-Ro tasks are 65536 and 16384, respectively. The total updates are 50K, 50K and 100K for *base*, *deep* and *big* in En-De task, respectively. We adopt Adam (Kingma & Ba, 2015) as an optimizer with an adam_β of (0.9, 0.997). The learning rate scheduler is *invert sqrt* with a learning rate of 0.002 and warmup updates of 16000. We also adopt label smoothing with a ratio of 0.1 in all the experiments. More details are exhibited in Table 11. During the evaluation process, we set the beam number to 4 and the length penalty to 0.6 for the En-De task. As for the En-Ro task, the number of beams is 5 and the length penalty is 1.3.

Table 9: The details of datasets.

Dataset	**Sentence**			**BPE**	**Vocab**
	Train	**Dev**	**Test**		
WMT'14 En-De	4.5M	3.0K	3.0K	32K	34040
WMT'16 En-Ro	0.6M	2.0K	2.0K	20K	19064
CNN/DailyMail	287K	13.0K	11.0K	30K	32584
CONLL	827K	5.4K	1.3K	30K	33136

A.2 ABSTRACTIVE SUMMARIZATION TASK

Dataset For abstractive summarization, we conduct experiments on a widely used corpus, e.g., CNN/DailyMail dataset. It consists of 287K training documents. Shared BPE operations with a size of 30K are performed on all the training data, resulting in a vocabulary of 32584.

Model Configuration We only provide the *base* configuration of our EIT and E-EIT for abstractive summarization. The details are presented in Table 10.

Training & Evaluation We train models for an abstractive summarization task on 8 GEFORCE RTX 3090 cards with a batch size of 131072 and total updates of 30K. We adopt a weight decay strategy with a ratio of 0.0001. Other hyper-parameters are the same as that in machine translation tasks. You can find their settings in Table 11. During testing, the number of beams is set to 4 and the length penalty is set to 2.0. Besides, we set the minimal length and maximum length to 55 and 140, respectively.

Table 10: The configurations of models on three sequence generation tasks. MT, AS and GEC denote machine translation, abstractive summarization and grammar error correction, respectively.

Task	Model	Configuration	M	$\mathbf{M^H}$	$\mathbf{M^{H_{isi}}}$	$\mathbf{M^{H_{csi}}}$	r	$\mathbf{K}_h^\psi$	$\mathbf{K}_w^\psi$	$\mathbf{K}_h^\phi$	$\mathbf{K}_w^\phi$
MT	EIT	*base*	8	8	128	64	8	1	7	1	3
		deep	8	8	128	64	8	1	7	1	3
		big	16	16	256	256	16	1	7	1	3
	E-EIT	*base*	8	32	-	-	8	1	7	1	7
		deep	8	32	-	-	8	1	7	1	7
		big	16	64	-	-	16	1	7	1	7
AS	EIT	*base*	8	8	8	64	8	1	1	1	1
	E-EIT	*base*	8	16	-	-	8	1	1	1	1
GEC	EIT	*base*	8	8	128	128	8	1	7	1	3
	E-EIT	*base*	8	64	-	-	8	1	7	1	7

Table 11: The training setups of different tasks. UF denotes the update frequency of the gradient. (.) lists the values of hyper-parameters under the *big* configuration, which vary from the values under the *base* configuration.

Hyper-parameter	WMT'14 En-De	WMT'16 En-Ro	CNN/DailyMail	CONLL
GPUs	8	4	8	8
Batch	4096	4096	4096	4096
UF	2	1	4	2
Optimer	Adam	Adam	Adam	Adam
Adam_β	(0.9, 0.997)	(0.9, 0.997)	(0.9, 0.997)	(0.9, 0.980)
LR	0.0020	0.0020	0.0020	0.0015
LR scheduler	inverse sqrt	inverse sqrt	inverse sqrt	inverse sqrt
Initial LR	$1e^{-7}$	$1e^{-7}$	$1e^{-7}$	$1e^{-7}$
Total updates	50K (100K)	25K	30K	14K
Warmup updates	16000	8000	8000	4000
Weight decay	0.0000	0.0000	0.0001	0.0001
Label smoothing	0.1	0.1	0.1	0.1
Dropout	0.1 (0.3)	0.1 (0.3)	0.1	0.2
Attention dropout	0.1	0.1	0.1	0.1
ReLU dropout	0.1	0.1	0.1	0.1
Word dropout	0.0	0.0	0.0	0.2

A.3 GRAMMAR ERROR CORRECTION TASK

Dataset For the grammar error correction task, we select the CONLL dataset to evaluate our approach. The CONLL dataset consists of 827K training sentences. We replicate the setup in Chollampatt & Ng (2018) and adopt the word-level dropout technique (Sennrich et al., 2016) to alleviate the overfitting problem. More details are listed in Table 9.

Model Configuration For grammar error correction task, we only provide the *base* configuration of our EIT and E-EIT. The details are presented in Table 10. Notice that the models on this task adopt a post-normalization strategy.

Training & Evaluation We train models for the grammar error correction task on 8 GEFORCE RTX 3090 cards. The batch size is 65536 and the total updates are 14K. More training details are shown in Table 11. During testing, the beams and length penalty are set to 6 and 0.6, respectively.

A.4 AUTOMATIC DISEASE DIAGNOSIS TASK

Dataset For the automatic disease diagnosis task, we select the ABIDE dataset to evaluate our approach. The ABIDE dataset consists of 1009 brain networks from 1009 real samples of 17 international sits. Due to the heterogeneity of this data, we adopt the shared data with re-standardized data splitting in Kan et al. (2022). Specifically, 70%, 10% and 20% samples are served as the training, validation and test sets, respectively.

Model Configuration For ABIDE task, we still follow the model configuration in Kan et al. (2022). Specifically, we build our BrainNetEITF with two-layer encoder. The number of heads M are set to 4 for each layer.

Training & Evaluation We train all models including the BrainNetTF and BrainNetEITF fro 200 epochs on a single GEFORCE RTX 3090 card. Each model is trained by 5 times. We adopt Adam (Kingma & Ba, 2015) as an optimizer with an initial learning rate of 10^{-4} and a weight decay of 10^{-4}. The bath size is set to 64. We adopt the checkpoint of the final epoch is used for evaluating the test set.

A.5 LANGUAGE MODELING TASK

Dataset For the language modeling task, we select the WikiText-103 dataset to evaluate our approach. The training set consists of 103M words from 28K articles. While for the validation and test sets, they are made up of 218K and 246K words, respectively. In details, we follow the instructions in Fairseq (Ott et al., 2019) to obtain and preprocess the data.

Model Configuration For WikiText-103 task, Both baseline and our model are all 8-layer big model with 8 heads. Note that the baseline we adopted are adaptive input transformer (Baevski & Auli, 2019). In this task, the kernel sizes in DEI are all set to 1.

Training & Evaluation The training and evaluation settings are all follow the standard instructions for language modeling in PyTorch (Ott et al., 2019). Note that due to the limit time of rebuttal, we only train both baseline and EIT with 169692 updates. All these configurations can be easily found in your fairseq codebase.

B DETAILS OF ANALYSIS

B.1 CALCULATION OF HEAD DISTANCE

Inspired by the attention metrics in Zhou et al. (2021a) and Wang et al. (2022), we measure the distance between different subspaces by calculating cosine similarity among attention maps. Notice that our metric focuses on the diversity of attention maps, which is quite different from them. Given two attention maps $\mathbf{A}^1$, $\mathbf{A}^2 \in \mathbb{R}^{T\times T}$, $\mathrm{Sim}(\mathbf{A}^1, \mathbf{A}^2)$ is calculated as follows:

$$\mathrm{Sim}(\mathbf{A}^1, \mathbf{A}^2) = \frac{1}{T}\sum_{i=1}^{T}\frac{|\mathbf{A}^1_{i,:}\mathbf{A}^2_{i,:}|}{\|\mathbf{A}^1_{i,:}\|_2\|\mathbf{A}^2_{i,:}\|_2} \tag{6}$$

Then we can obtain the head similarity by applying Eq. (6) to attention maps from every two subspaces and average them:

$$\mathrm{Sim}_{\mathrm{head}}(\mathbf{A}) = \frac{1}{M(M-1)}(\sum_{i=1}^{M}\sum_{j=1}^{M}\mathrm{Sim}(\mathbf{A}^i, \mathbf{A}^j) - M) \tag{7}$$

The obtained head similarity ranges from [0, 1]. The larger the head similarity is, the lower the distances between different subspaces are.

Table 12: Detailed parameters of our methods on WMT En-De and WMT En-Ro tasks.

Model	En-De			En-Ro		
	Base	Deep-48L	Big	Base	Deep-24L	Big
Transformer (Pre)	61.56M	193.96M	211.22M	53.90M	110.64M	195.88M
EIT	61.63M	194.32M	211.55M	53.98M	111.09M	196.40M
E-EIT	61.57M	194.14M	211.30M	53.92M	110.73M	195.97M

B.2 CALCULATION OF TOKEN CORRELATION

We define a metric $\mathcal{TC}$, which measures the correlation among the representations of different tokens. Given a sequence representation $X \in \mathbb{R}^{T\times d}$, which consists of T tokens, the token correlation of X can be formally defined as:

$$\mathcal{TC}(X) = \frac{1}{T(T-1)} \sum_{1\le i,j\le T \wedge j\neq i} \rho(X_{i,\cdot}, X_{j,\cdot}) \tag{8}$$

where $\rho(\cdot)$ is the pearson correlation coefficient (Benesty et al., 2009), which measures the correlation between two vectors:

$$\rho(\mathbf{x},\mathbf{y}) = \frac{\sum_1^d (x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^d (x_i-\bar{x})^2 \sum_{i=1}^d (y_i-\bar{y})^2}} \tag{9}$$

Intuitively, the larger the $\mathcal{TC}$ is, the higher the token correlation is, degrading the model's learning capacity (Gong et al., 2021).

B.3 CALCULATION OF UTILIZATION RATIO OF FUNCTIONS

Inspired by Liu et al. (2020a), we define a metric, namely $\mathcal{UR}$, to measure the utilization ratio of functions. Given a function f, its utilization ratio $\mathcal{UR}(f)$ is calculated as follows:

$$\mathcal{UR}(f) = \frac{\mathrm{std}(f)}{\mathrm{std}(f+res)} \tag{10}$$

where $\mathrm{std}(f)$ denotes the standard deviation of the output of function f and $\mathrm{std}(f+res)$ denotes the standard deviation of the output of function f added the residual.

C VISUALIZATION OF TRAINING AND VALIDATION PERPLEXITY

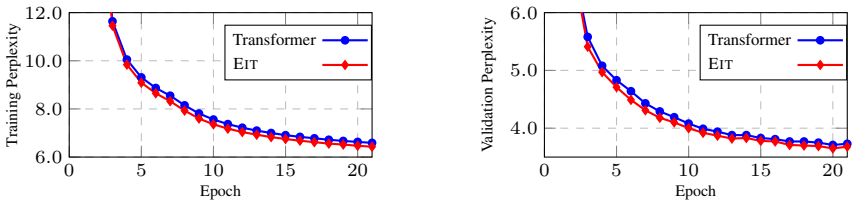


Figure 9: Training perplexity and validation perplexity of Transformer and our EIT on WMT'14 En-De task. Note that the models are in *base* configuration.

We plot the training and validation perplexity of Transformer and our EIT on the WMT'14 task in Figure 9. We can see that our EIT owns lower training and validation perplexity than Transformer.

D DETAILED ADDED PARAMETERS OF OUR METHODS

The detailed parameters of our methods are listed in Table 12.

Table 13: Effect on placement of DEI on two tasks. Here, IB and IA denote the ISI module is located *before* or *after* the Softmax function, and so on for CSI.

#	IB	IA	CB	CA	En-De	En-Ro
1	✓		✓		**28.00**	**35.10**
2	✓			✓	27.93	34.88
3		✓		✓	27.50	34.45

E THEORETICAL COMPLEXITY ANALYSIS OF GROUP CONVOLUTION

Given an input representation, namely $\mathbf{H} \in \mathbb{R}^{d_i \times s_h \times s_w}$, we process it with a standard convolution and a group convolution to obtain the output representations $\hat{\mathbf{H}}^s \in \mathbb{R}^{d_o \times s_h^o \times s_w^o}$ and $\hat{\mathbf{H}}^g \in \mathbb{R}^{d_o \times s_h^o \times s_w^o}$, respectively, where d_i is the dimension of the input feature, d_o is the dimension of the output feature, s_h is the height of the input feature, s_w is the width of the input feature, s_h^o is the height of the output feature and s_w^o is the width of the output feature.

When processing by a group convolutions with an output dimension d_g^o, group g and a kernel size (K_h, K_w), we first reshape $\mathbf{H}$ to obtain $\mathbf{H}^g \in \mathbb{R}^{g \times d_g \times s_h \times s_w}$ and generate the corresponding output, namely $\hat{\mathbf{H}}^g \in \mathbb{R}^{d_o \times s_h^o \times s_w^o}$, after reshaping the original output $\hat{\mathbf{H}}^{g_m} \in \mathbb{R}^{g \times d_g^o \times s_h^o \times s_w^o}$, where g is the number of groups, d_g, equal to $\frac{d_i}{g}$, is the dimension of each group in the input feature and d_g^o equals $\frac{d_o}{g}$ which denotes the dimension of each group in the output feature. During this process, we can obtain that the number of its parameters, namely N_g, is $d_g^o \times d_g \times g \times K_h \times K_w$ and its computational complexity, namely C_g, is $O(g \times d_g^o \times s_h^o \times s_w^o \times K_h \times K_w \times d_g)$.

Correspondingly, when processing $\mathbf{H}$ by a standard convolution with an output dimension d_o and a kernel size (K_h, K_w), the number of its parameters, namely N_s, is $d_o \times d_i \times K_h \times K_w$ and its computational complexity, namely C_s, is $O(d_o \times s_h^o \times s_w^o \times K_h \times K_w \times d_i)$. By simplifying these expressions, we can get the final equations as follows: $N_g = \frac{d_o}{g} \times d_i \times K_h \times K_w = \frac{N_s}{g}$, $C_g = O(\frac{d_o}{g} \times d_i \times s_h^o \times s_w^o \times K_h \times K_w) \approx \frac{C_s}{g}$, which demonstrates that the group convolutions have theoretically computational efficiency and parametric efficiency.

F ANALYSIS ON PLACEMENT OF DEI

Table 13 compares the impacts on several placements of DEI module, e.g., ISI→ Softmax →CSI. First, Softmax operation is insensitive to the placement of the CSI sub-module, which results in negligible BLEU degradation (#1 vs. #2). Moreover, by comparing #2 and #3, we find that when placing the ISI sub-module behind the Softmax, the performance suffers a dramatic BLEU drop on both tasks. A possible explanation is that the ISI module destroys the normalized attributes of the attention distributions since the convolutions with large kernel sizes introduce more noise in the original distributions. This also indicates that a soft rectification of attention distributions can boost the quality of attention distributions, which highly correlates with the findings of previous work Zhou et al. (2021b).

G WHETHER AND HOW MUCH EIT VARIANTS BENEFIT FROM ENLARGED SUBSPACES

Intuitively, EIT can benefit from more heads under the same configuration, as it adopts the *many-to-many* scheme. To validate this, we select only one attention map and share it across all the heads. To be more specific, we still follow the calculation process to obtain the attention maps. The main difference is that we modify the Eq. (1) as follows: $\mathbf{O} = \left[\mathbf{O}^1, \ldots, \mathbf{O}^M\right] \mathbf{W}_O = \left[\mathbf{A}^1\mathbf{V}^1, \ldots, \mathbf{A}^1\mathbf{V}^M\right] \mathbf{W}_O$. The results are exhibited in Table 14. We find that both EIT and E-EIT with selected attention map are still superior to the standard Transformer on two translation

Table 14: Ablation study on the expressiveness of subspaces.

#	Model	En-De	En-Ro
1	Transformer	27.13	34.23
2	EIT (selected attn)	**27.20**	**34.72**
3	E-EIT (selected attn)	27.17	34.65

tasks. This indicates that the expressiveness of one head in EMHA is comparable to that of eight heads in standard MHSA, which also demonstrates EIT and E-EIT have high efficiency in attaining gains from the enlarged subspaces.

H EFFECT OF SIZE OF RECEPTIVE FIELD AND NUMBER OF HEADS

We also investigate the connection between performance and two important hyper-parameters: *Respective Field* and *Number of Heads*. We first give a general definition of receptive field with any size r as follows: $\Psi(i, r, U_K) = \{\mathbf{K}^k | \mathbf{K}^k \in U_K \wedge k = (i + j - 1)\%M, j = 1, 2, \ldots, r\}$. With such definition, we conduct experiments with different r and different M. The results are exhibited in Figure 10. Intuitively, the larger the size of the receptive field and the number of heads, the more attention maps we can obtain. We observe that the performance gain increases as the sizes of these parameters grow. This reveals that EIT can indeed benefit from more attention maps.

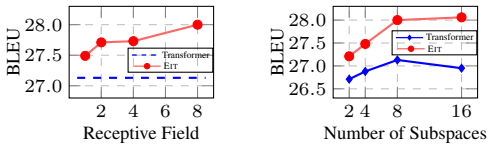


Figure 10: BLEU vs. receptive field (left) and the number of subspaces (right).

I EFFICIENCY/PERFORMANCE TRADE-OFFS

I.1 NUMBER OF EIT ENCODER LAYERS

We further dig out the effect of the number of EIT encoder layers on the performance. This can also be served as a way to trade-off between computational efficiency and performance. Figure 11 exhibited the results. We can see our EIT can maintain a high-level performance even when the number of EIT layers is one. The reason is that, as illustrated in Figure6(c), the standard Transformer has a higher token correlation even in the first layer, which does harm to the learning of later layers, while our EIT in the shallow layers can maintain a low level token correlation. Thus, they can collaborate well.

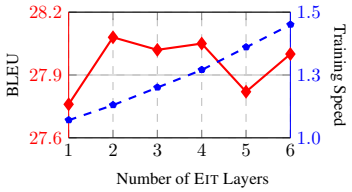


Figure 11: Efficiency/Performance trade-offs against EIT layers

Motivated by this phenomenon, we further apply such design to the big configuration. We set the number of layers as 2, and keep the training strategy the same as that in Appendix A. Surprisingly, We can train a big model with a performance of 29.29 within 7.12 hours, which cut off 31% training costs while improving 1.7% performance.

I.2 OTHER EFFICIENCY/PERFORMANCE TRADE-OFFS

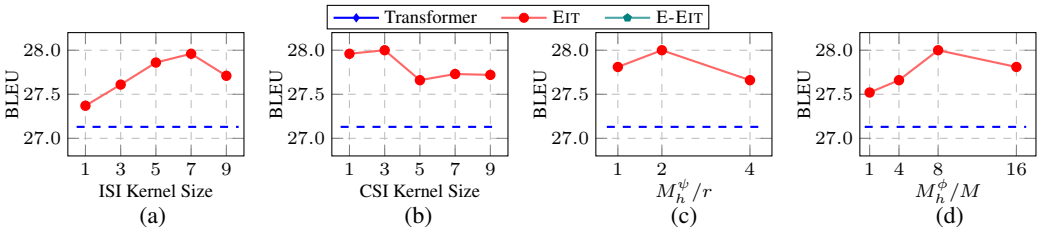


Figure 12: The comparison of BLEU against different hyper-parameters. Note that the blue horizontal line represents the performance of Transformer.

Since there are several hyper-parameters in both ISI and CSI sub-modules, it is necessary to figure out how they affect performance. Figure 12 (a-d) plots the performance of EIT against the kernel

size and the increasing rate. We find that the ISI sub-module prefers a larger kernel size than the CSI sub-module. A reliable explanation is that extracting the representative information from sufficient attention maps requires a large receptive field. Moreover, we find that EIT achieves the best performance with the M_h^ψ and M_h^ϕ of $2r$ and $8M$, respectively. Here M_h^ψ and M_h^ϕ represent the intermediate head size in ISI and CSI, respectively. These observations further help us trade off efficiency and performance well.

J IS PERFORMANCE HIGHLY RELATED TO DIVERSITY OF ATTENTION MAPS?

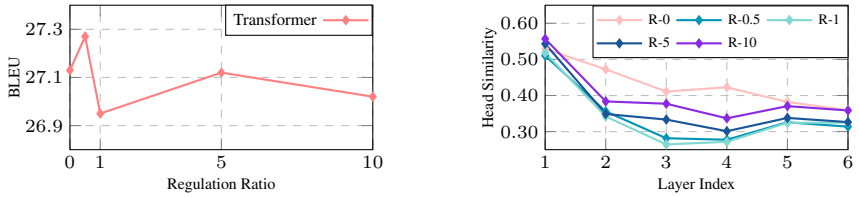


Figure 13: The performance of Transformer and its corresponding head similarity, against different ratios of regulation. The selected regulation item is the same as the position regulation item in Li et al. (2018). R-X represents the model with regulation item ratio of X.

We conduct detailed experiments on the WMT'14 En-De translation task to answer this question. We exhibit the results in Figure 13, where we show the performance of Transformer with different ratios of restriction (left) as well as their corresponding similarity among attention maps learnt from different heads (right). In most cases, models with this restraint are inferior to the standard Transformer. Moreover, the level of diversity among attention maps is not related to performance improvement. For instance, the R-1 model owning the lowest head similarity, achieves the lowest performance. All these indicate that similar attention positions are not a drawback for MHSA. In contrast, similar attention maps may result from learning universal features. All heads learn the most critical information for tasks, which results in high similarity. This hypothesis can also be validated by Figure 8(below) and Figure 18.

K MORE DISCUSSION ABOUT OVERSMOOTHING AND FEATURES COLLAPSE

We plot the cross-layer similarity of features in Figure 14. We follow the metric defined in Zhou et al. (2021a) to calculate the cross-layer similarity of features. We can see that the final representation learnt by EIT encoder is more distinct from that learnt by the first layer of encoder than Transformer. According to the findings in (Zhou et al., 2021a), we can conclude that EIT can extract more expressive representation. Moreover, combining Figure 14 with Figure 6(middle & right), we can observe following trends:

- As the token correlation increases, the utilization of function like attention or FFN decreases.
- As the utilization of function like attention and FFN decreases, the slop of cross-layer similarity curve becomes to flat.

All these seem to reveal that the *feature collapse* problem attributes to the oversmoothing of functions to some degree.

L MORE VISUALIZATION

Detailed Visualizations of Attention Maps In this section, we visualize attention maps of each period, e.g., RFE, ISI and CSI. The results are exhibited in Figure 15. We can see that RFE can indeed generate a large number of diverse attention maps. We also plot the attention maps generated by *Attention Expansion* in Figure 16. The attention maps generated by *Attention Expansion* own high similarity. Some attention maps even degrade to a nearly uniform distribution, such as the attention map placed in row 1 and column 5. These attention maps may lead to some inefficiency since they seem to be useless.

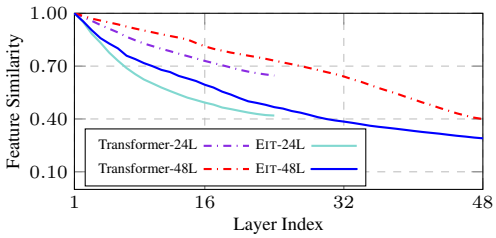
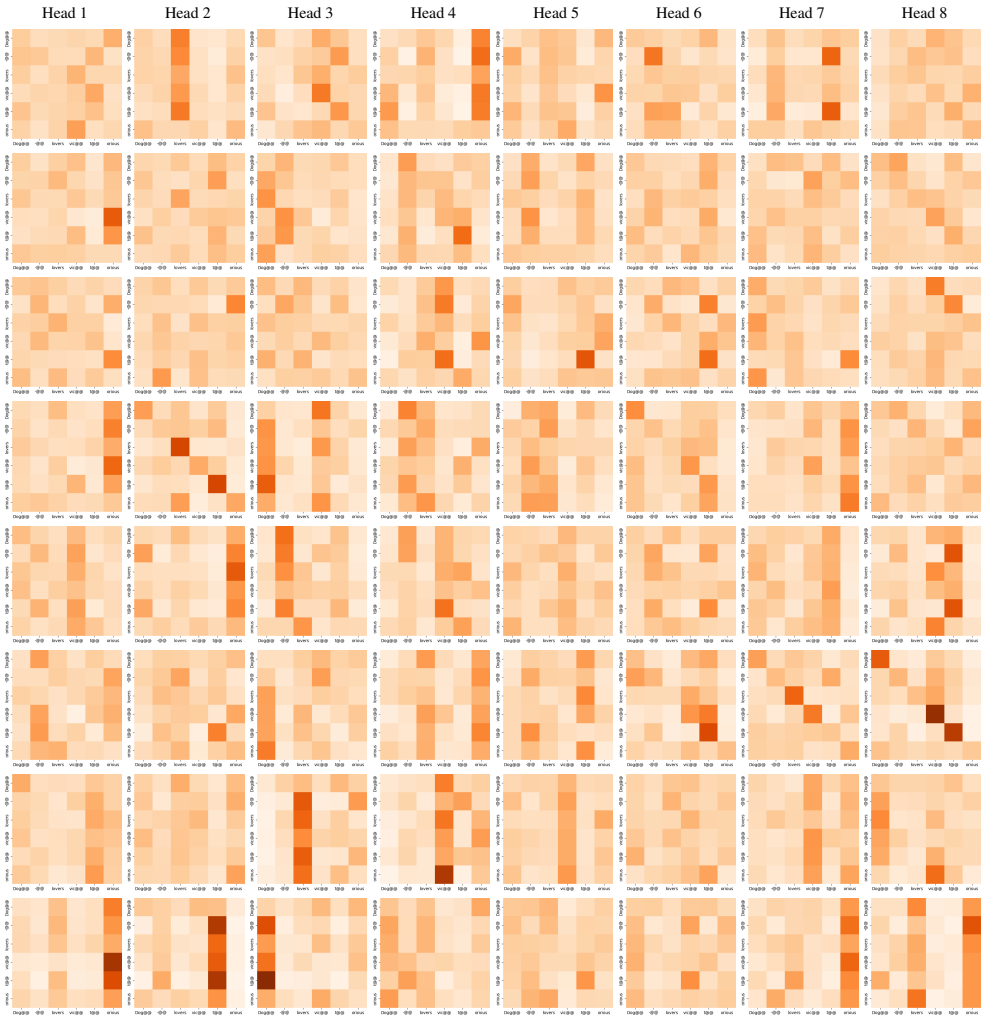


Figure 14: Quantitative analysis on feature similarity across layers. The features similarity is measured by the similarity between features of each layer and the features of first layer. We adopt the cosine similarity to measure the similarity (Zhou et al., 2021a).

Visualization of Examples in Abstractive Summarization Task We also plot the attention distributions of a real case in the abstractive summarization task in Figure 17 (Transformer) and Figure 18 (EIT). We can see that the attention maps of all subspaces generated by EIT have a strong weight within the diagonal region. As for Transformer, only a few attention maps can capture such information.



(a) After RFE

(b) After ISI

(c) After CSI

Figure 15: Quantitative examples of the attention distribution over a real case in the En-De task. We trace the attention maps along the whole calculation process. Dark color means a higher value in the distribution.

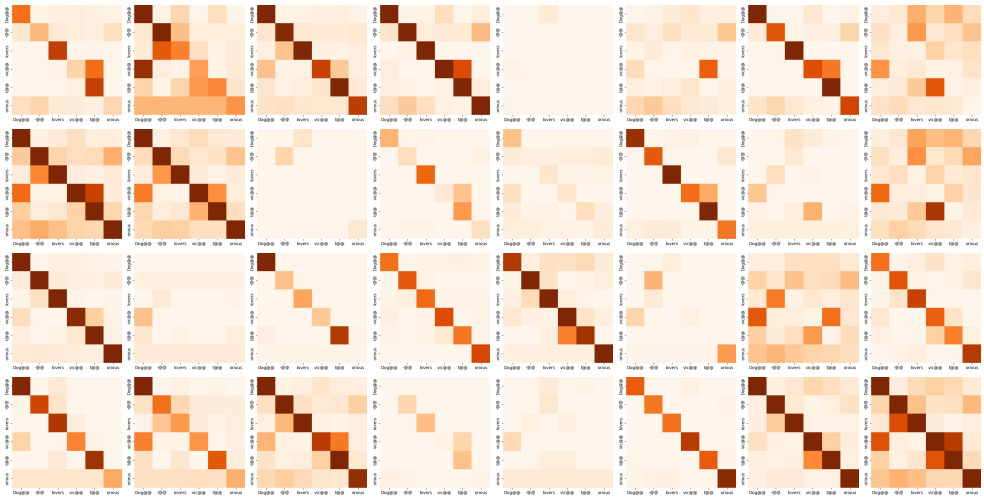


Figure 16: Quantitative examples of the attention distribution obtained by *Refiner* over a real case in the En-De task. Dark color means a higher value in the distribution. Note that we select the best setup (32) for *Refiner* to visualize.

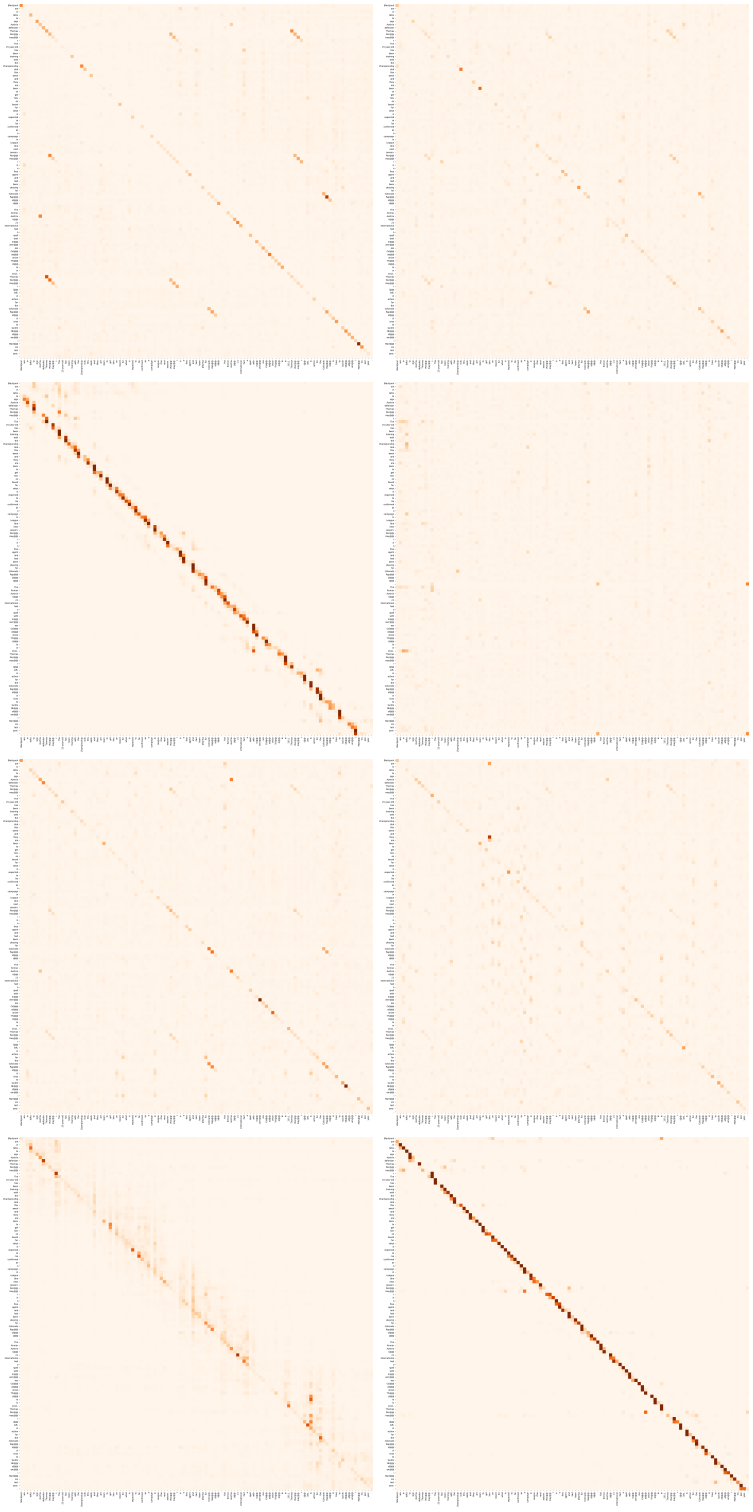


Figure 17: Quantitative examples of the attention distribution over a real case in the abstractive summarization task. The attention maps come from eight subspaces of Transformer. Dark color means a higher value in the distribution.

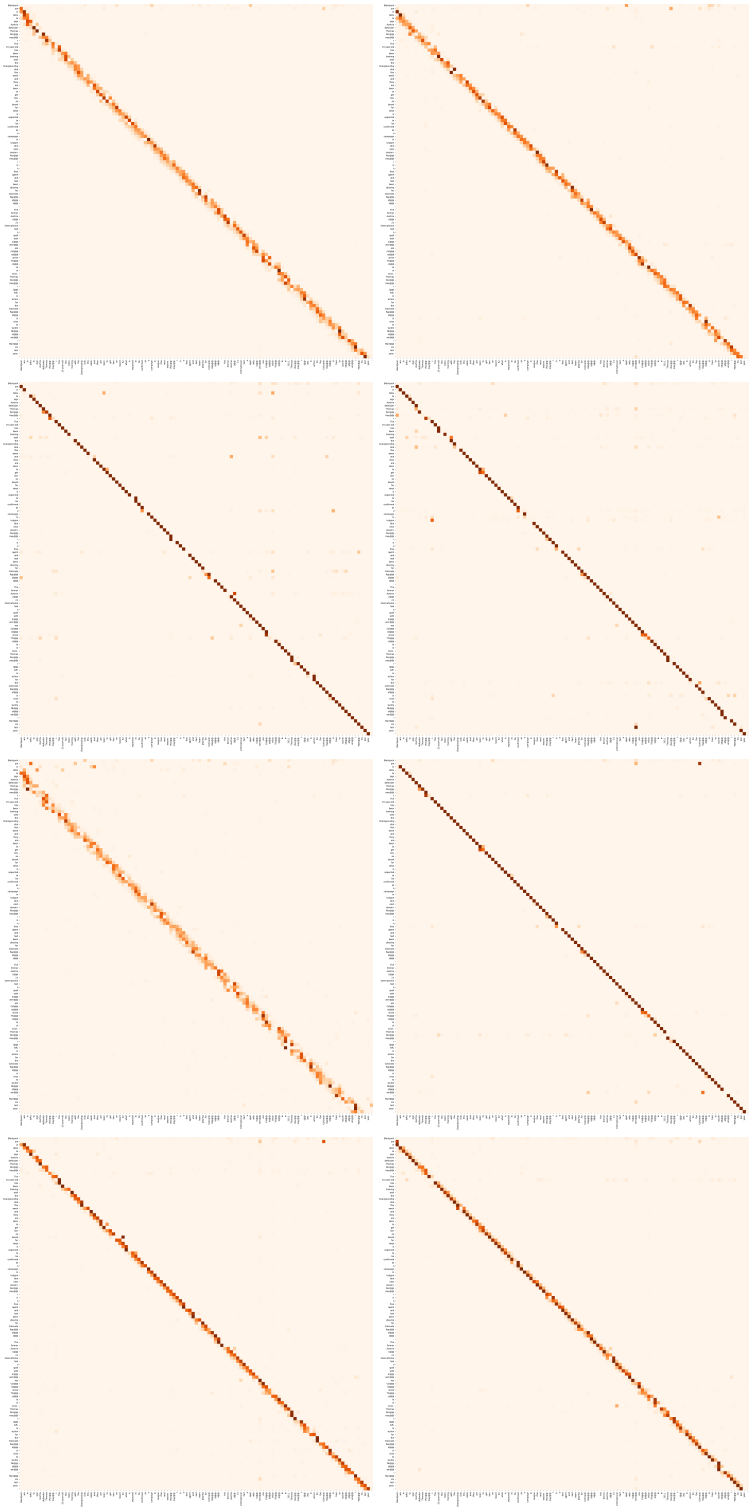


Figure 18: Quantitative examples of the attention distribution over a real case in the abstractive summarization task. The attention maps come from eight subspaces of EIT. Dark color means a higher value in the distribution.