

# Modeling Concentrated Cross-Attention for Neural Machine Translation with Gaussian Mixture Model

Shaolei Zhang<sup>1,2</sup>, Yang Feng<sup>1,2\*</sup>

<sup>1</sup>Key Laboratory of Intelligent Information Processing  
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)

<sup>2</sup>University of Chinese Academy of Sciences, Beijing, China  
{zhangshaolei20z, fengyang}@ict.ac.cn

## Abstract

Cross-attention is an important component of neural machine translation (NMT), which is always realized by dot-product attention in previous methods. However, dot-product attention only considers the pair-wise correlation between words, resulting in dispersion when dealing with long sentences and neglect of source neighboring relationships. Inspired by linguistics, the above issues are caused by ignoring a type of cross-attention, called concentrated attention, which focuses on several central words and then spreads around them. In this work, we apply Gaussian Mixture Model (GMM) to model the concentrated attention in cross-attention. Experiments and analyses we conducted on three datasets show that the proposed method outperforms the baseline and has significant improvement on alignment quality, N-gram accuracy, and long sentence translation.

## 1 Introduction

Recently, Neural Machine Translation (NMT) has been greatly improved with Transformer (Vaswani et al., 2017), which mainly relies on the attention mechanism. The attention mechanism in Transformer consists of self-attention and cross-attention, where cross-attention is proved more important to translation quality than self-attention (He et al., 2020; You et al., 2020). Even if the self-attention is modified to a fixed template, the translation quality would not significantly reduce (You et al., 2020), while cross-attention plays an irreplaceable role in NMT. Cross-attention in Transformer is realized by dot-product attention, which calculates the attention distribution base on the pair-wise similarity.

However, modeling cross-attention with the dot-product attention still has some weaknesses due to its calculation way. First, when dealing with

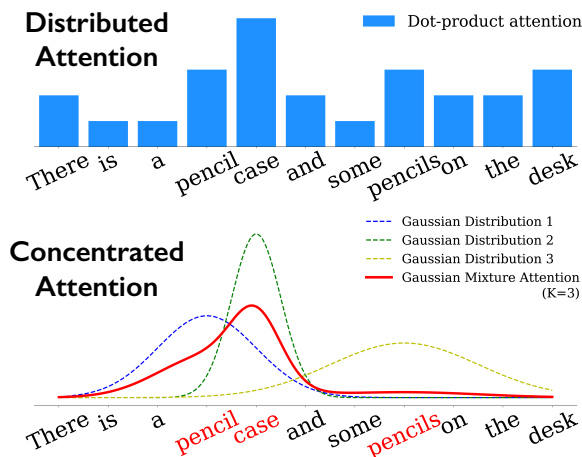


Figure 1: An attention example of En→De translation when generating target words “Mäppchen” (English meaning: pencil case), showing the difference and complementarity between distributed attention and concentrated attention.

long sentences, the attention distribution with dot-product attention tends to be dispersed (Yang et al., 2018), which proved unfavorable for translation (Zhang et al., 2019; Tang et al., 2019; He et al., 2020). Second, dot-product attention is difficult to explicitly consider the source neighboring relationship (Im and Cho, 2017; Sperber et al., 2018), resulting in ignoring the words with lower similarity but nearby the important word which determine phrase structure or morphology.

Research in linguistics and cognitive science suggests that human attention to language can be divided into two categories: *distributed attention* and *concentrated attention* (Jacob and Bruce, 1973; Ito et al., 1998; Brand and Johnson, 2018). Specifically, distributed attention is scattered on all source words, and the degree of attention is determined by correlation. On the contrary, concentrated attention only focuses on a few central words and then spreads on the words around them. Accordingly, we consider that cross-attention can be divided into these two types of attention as well, where dis-

\* Corresponding author: Yang Feng.

tributed attention is well modeled by dot-product attention, but concentrated attention is ignored. Figure 1 shows an attention example of En→De translation when generating target words “Mäppchen” (English meaning: pencil case). Only with distributed attention, some irrelevant words (such as ‘desk’) robbed some attention weight, resulting in attention dispersion. Besides, the correlation to the function words (both ‘a’ and ‘some’) are low and similar, but they are important to determine the singular/plural forms of the target word. In concentrated attention, attention distribution is more concentrated and can capture neighboring relationships, which make up for the lack of distributed attention.

In this paper, to explicitly model the concentrated cross-attention, we apply the Gaussian mixture model (GMM) (Rasmussen, 1999) to construct *Gaussian mixture attention*. Specifically, Gaussian mixture attention first focuses on some central words, and then pays attention to the words around the central words, where the attention decreases as the word away from the central word. Since cross-lingual alignments are often one-to-many, Gaussian mixture attention is more flexible to model multiple central words, which is not possible with a single Gaussian distribution. Then, Gaussian mixture attention and dot-product attention are fused to jointly determine the total attention.

Experiments we conducted on three datasets show that our method outperforms the baseline on translation quality. Further analyses show that our method enhances cross-attention, thereby improving the performance of alignment quality, N-gram accuracy and long sentence translation.

Our contributions can be summarized as follows:

- We introduce concentrated attention to cross-attention, which successfully compensates for the weakness of dot-product attention.
- To our best knowledge, we are the first to apply GMM to model attention distribution in text sequence, which provides a method for modeling multi-center attention distribution.

## 2 Background

Our method is applied on cross-attention in Transformer (Vaswani et al., 2017), so we first briefly introduce the architecture of Transformer with a focus on its dot-product attention. Then, we give the concept of the Gaussian Mixture model.

### 2.1 Transformer

**Encoder-Decoder** Transformer consists of encoder and decoder, each of which contains  $N$  repeated independent structures. Each encoder layer contains two sub-layers: self-attention and fully connected feed-forward network (FFN), while each decoder layer includes three sub-layers: self-attention, cross-attention, and FFN. We denote the input sentence as  $\mathbf{x} = (x_1, \dots, x_J)$ , where  $J$  is the length of source sentence,  $x_j \in \mathbb{R}^{d_{model}}$  is the sum of the token embedding and the position encoding of the source token, and  $d_{model}$  represents the representation dimension. The encoder maps  $\mathbf{x}$  to a sequence of hidden states  $\mathbf{z} = (z_1, \dots, z_J)$ . Given  $\mathbf{z}$  and previous target tokens, the decoder predicts the next output token  $y_i$ , and the entire output sequence is  $\mathbf{y} = (y_1, \dots, y_I)$ , where  $I$  is the length of the target sentence.

**Dot-product attention** Both self-attention and cross-attention in Transformer apply multi-head attention, which contains multiple heads and each head accomplishes scaled dot-product attention to process a set of queries( $q$ ), keys( $k$ ), and values( $v$ ). Following, we focus on the specific representation of the dot-product attention in cross-attention.

In cross-attention, the queries is the hidden states of decoder  $\mathbf{s} = \{s_1, \dots, s_I\}$ , while the keys and values both come from the hidden states of the encoder  $\mathbf{z} = \{z_1, \dots, z_J\}$ . Dot-product attention first calculates the pairwise correlation score  $e_{ij}$  between the  $i^{th}$  target token and the  $j^{th}$  source token, and normalizes to obtain the dot-product attention weight  $\alpha_{ij}$  of the  $i^{th}$  target token for each source token:

$$e_{ij} = \frac{Q(s_i)K(z_j)^T}{\sqrt{d_k}} \quad (1)$$

$$\alpha_{ij} = \frac{\exp e_{ij}}{\sum_{l=1}^J \exp e_{il}} \quad (2)$$

where  $Q(\cdot)$  and  $K(\cdot)$  are the projection functions from the input space to the query space and the key space, respectively, and  $d_k$  represents the dimensions of the queries and keys. Then for each  $i$ , the context vector  $c_i$  is calculated as:

$$c_i = \sum_{j=1}^n \alpha_{ij}V(z_j) \quad (3)$$

where  $V(\cdot)$  is a projection function from the input space to the value space.

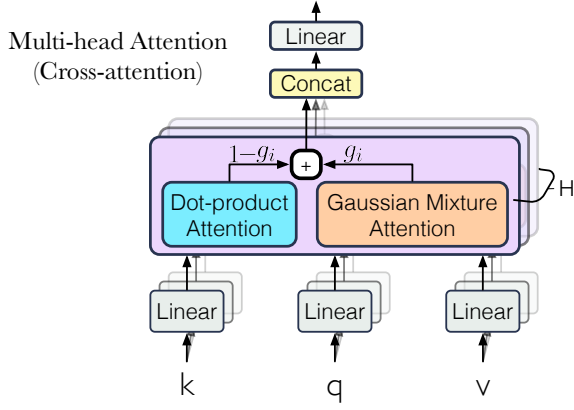


Figure 2: The architecture of the proposed method.

## 2.2 Gaussian Mixture Model

**Single Gaussian distribution** A Gaussian distribution with the mean  $\mu$  and variance  $\sigma$ , which is calculated as:

$$x \sim \mathcal{N}(\mu, \sigma) \equiv \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \quad (4)$$

**Gaussian mixture model (GMM)** (Rasmussen, 1999) Composed of  $K$  single Gaussian distribution, which is calculated as:

$$x \sim \sum_{k=1}^K a_k \mathcal{N}(\mu_k, \sigma_k) \quad (5)$$

where  $a_k$ ,  $\mu_k$  and  $\sigma_k$  are weight, mean and variance of the  $k^{th}$  Gaussian distribution, respectively. During training, for unlabeled data, GMM can be trained using the EM algorithm, and for labeled data, GMM can be trained using methods such as maximum likelihood or gradient descent.

## 3 The Proposed Method

To improve cross-attention, in addition to dot-product attention, we introduce the Gaussian mixture attention into each head of cross-attention. As shown in Figure 2, we first calculate the dot-product attention and Gaussian mixture attention, and then fuses them through a gating mechanism to determine the total attention distribution. The proposed Gaussian mixture attention is constructed by mean, variance, and weight, all of which are predicted based on the target word, as shown in Figure 3. The specific details will be introduced following.

### 3.1 Gaussian Mixture Attention

Gaussian mixture attention consists of  $K$  independent Gaussian distributions, where each Gaussian

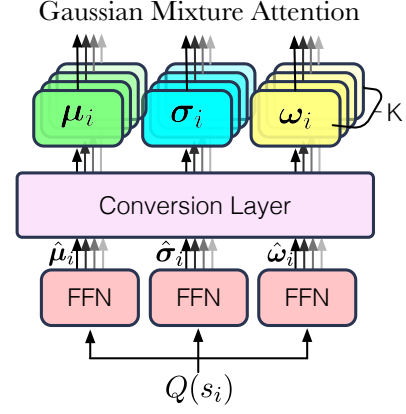


Figure 3: Calculation of Gaussian Mixture Attention.

distribution has different weights. The general form of Gaussian mixture attention  $\beta_{ij}$  between the  $i^{th}$  target token and the  $j^{th}$  source token is defined as:

$$\beta_{ij} = \sum_{k=1}^K \omega_{ik} \cdot \frac{1}{Z_{ik}} \exp\left(-\frac{(j-\mu_{ik})^2}{2\sigma_{ik}^2}\right) \quad (6)$$

where  $\mu_{ik}$ ,  $\sigma_{ik}$  and  $\omega_{ik}$  are the mean, variance, and the weight of the  $k^{th}$  Gaussian distribution of the  $i^{th}$  target word, respectively,  $Z_{ik}$  represents the normalization factor of  $k^{th}$  Gaussian distribution of the  $i^{th}$  target word, and  $K$  is a hyperparameter we set. In the physical sense, the mean  $\mu_{ik}$  represents the position of the central word, the variance  $\sigma_{ik}$  represents the attenuation degree of attention with the offset from the central word, and the weight  $\omega_{ik}$  represents the importance of the central word.

As shown in Figure 3, the parameters for the Gaussian mixture attention of the  $i^{th}$  target word are  $(\omega_i, \mu_i, \sigma_i, Z_i)$ , where  $\omega_i \in \mathbb{R}^{K \times 1}$  is the vector representation of  $[\omega_{i1}, \dots, \omega_{iK}]$ , and others are in the same rule.  $(\omega_i, \mu_i, \sigma_i, Z_i)$  are converted from predicted intermediate parameters  $(\hat{\omega}_i, \hat{\mu}_i, \hat{\sigma}_i)$ . According to Yang et al. (2018) and Battenberg et al. (2020), it is more robust to use target hidden state to predict variables of Gaussian distribution. Thus, the intermediate parameters are predicted through the Feedforward Network (FFN):

$$\hat{\omega}_i = \mathbf{V}_\omega^\top \tanh\left(\mathbf{W}_\omega^\top Q(s_i) + b_{\omega 1}\right) + b_{\omega 2} \quad (7)$$

$$\hat{\mu}_i = \mathbf{V}_\mu^\top \tanh\left(\mathbf{W}_\mu^\top Q(s_i) + b_{\mu 1}\right) + b_{\mu 2} \quad (8)$$

$$\hat{\sigma}_i = \mathbf{V}_\sigma^\top \tanh\left(\mathbf{W}_\sigma^\top Q(s_i) + b_{\sigma 1}\right) + b_{\sigma 2} \quad (9)$$

where  $\{\mathbf{W}_\omega, \mathbf{W}_\mu, \mathbf{W}_\sigma\} \in \mathbb{R}^{d_q \times d_q}$  and  $\{\mathbf{V}_\omega, \mathbf{V}_\mu, \mathbf{V}_\sigma\} \in \mathbb{R}^{d_q \times K}$  are learnable parameters,  $\{b_{\omega 1}, b_{\mu 1}, b_{\sigma 1}\} \in \mathbb{R}^{d_q \times 1}$  and  $\{b_{\omega 2}, b_{\mu 2}, b_{\sigma 2}\} \in \mathbb{R}^{K \times 1}$  are learnable

bias. Note that  $Q(\cdot)$  shares parameters with the projection function of dot-product attention in Eq.(2), and parameters of FFN are shared in each head.

Given intermediate parameters  $(\hat{\omega}_i, \hat{\mu}_i, \hat{\sigma}_i)$ , our method predicts  $(\omega_i, \mu_i, \sigma_i, \mathbf{Z}_i)$  through a conversion layer. For the weight of every single Gaussian distribution, we normalize them with:

$$\omega_i = \text{Softmax}(\hat{\omega}_i) \quad (10)$$

For the mean, considering the word order differences between language, we directly predict its absolute position:

$$\mu_i = J \cdot \text{Sigmoid}(\hat{\mu}_i) \quad (11)$$

where  $J$  is the length of the source sentence.

Note that, since the source position is discrete and will be truncated at the boundary, the attention weight sum is less than 1 without normalization, rather than fully normalized attention weight. Previous work (Luong et al., 2015; Yang et al., 2018; You et al., 2020) on applying Gaussian distribution hardly normalized it since the normalization of Gaussian attention weights results in unstable training. However, unnormalized attention weight leads to occasional spikes or dropouts in the attention and alignment (Battenberg et al., 2020). Therefore, to normalize Gaussian mixture attention meanwhile maintaining training stability, we propose an approximate normalization.

Approximate normalization adjusts the variance according to the mean position to ensure that most of the weights are within the source sentence, which not only avoids the spikes caused by little weights sum but also ensures stable training. For approximate normalization, we calculate the value of  $\sigma_i$  with  $\mu_i$ :

$$\sigma_i = \min \left\{ \frac{J}{6} \cdot \text{Sigmoid}(\hat{\sigma}_i), \frac{\mu_i}{3}, \frac{J - \mu_i}{3} \right\} \quad (12)$$

Approximate normalization ensures that interval  $[\mu_i - 3\sigma_i, \mu_i + 3\sigma_i]$  is within the source sentence (Pukelsheim, 1994). Besides,  $\mathbf{Z}_i$  is set to  $\sqrt{2\pi\sigma_i^2}$  to normalize each Gaussian distribution in GMM. In general, although Gaussian mixture attention is not strictly normalized, approximate normalization guarantees a coverage of more than 90% of the attention weight. In the experiments (Sec.5.2), we additionally report the results of a strict normalization Gaussian mixture attention as a variant of our method for comparison.

### 3.2 Fusion of Attention

Dot-product attention ( $\alpha_{ij}$  in Eq.(2)) capture the distributed attention brought by the pair-wise similarity, while Gaussian mixture attention ( $\beta_{ij}$  in Eq.(6)) model the location-related concentrated attention. To balance two types of attention, we calculate the total attention weight  $\gamma_{ij}$  by fusing them through a gating mechanism:

$$\gamma_{ij} = (1 - g_i) \times \alpha_{ij} + g_i \times \beta_{ij} \quad (13)$$

where  $g_i$  is a gating factor, predicted through FFN:

$$g_i = \text{Sigmoid}(\mathbf{V}_g^\top \tanh(\mathbf{W}_g^\top Q(s_i) + b_{g1}) + b_{g2}) \quad (14)$$

where  $\mathbf{W}_g \in \mathbb{R}^{d_q \times d_q}$  and  $\mathbf{V}_g \in \mathbb{R}^{d_q \times 1}$  are learnable parameters,  $b_{g1} \in \mathbb{R}^{d_q \times 1}$  and  $b_{g2} \in \mathbb{R}$  are learnable bias. Finally, the context vector  $c_i$  in Eq.(3) is calculated as:

$$c_i = \sum_{j=1}^n \gamma_{ij} V(z_j) \quad (15)$$

## 4 Related Work

Attention mechanism is the most significant component of the Transformer (Vaswani et al., 2017) for Neural Machine Translation. Recently, some methods model the location in Transformer, most of which focus on self-attention.

Some methods improve the word representation. Transformer itself (Vaswani et al., 2017) introduced a position encoding to embed position information in word representation. Shaw et al. (2018) introduced relative position encoding in self-attention. Wang et al. (2019) enhanced self-attention with structural positions from the syntax dependencies. Ding et al. (2020a) utilized reordering information to learn position representation in self-attention.

Other methods directly consider location information in the attention mechanism, which is closely related to this work. Luong et al. (2015) proposed local attention, which only focuses on a small subset of the source positions. Yang et al. (2018) multiplied a learnable Gaussian bias to self-attention to model the local information. Song et al. (2018) accommodates some masks for self-attention to extract global/local information. Xu et al. (2019) propose a hybrid attention mechanism to dynamically leverage both local and global information. You et al. (2020) apply a hard-code Gaussian to replace dot-product attention in Transformer.



Models	$Z_i$	$\omega_i$	$\mu_i$	$\sigma_i$	Normalized
Synthesis Network	1	$\exp(\hat{\omega}_i)$	$\mu_{i-1} + \exp(\hat{\mu}_i)$	$\sqrt{\exp(-\hat{\sigma}_i)/2}$	None
Our Method	$\sqrt{2\pi\sigma_i^2}$	Softmax( $\hat{\omega}_i$ )	$J \cdot \text{Sigmoid}(\hat{\mu}_i)$	Eq.(12)	Approximate
Our Method+Norm.	$\sqrt{2\pi\sigma_i^2}$	Softmax( $\hat{\omega}_i$ )	$J \cdot \text{Sigmoid}(\hat{\mu}_i)$	$J \cdot \text{Sigmoid}(\hat{\sigma}_i)$	Strict

Table 1: Conversion method of synthesis network, our method, and the strict-normalized variant of our method.

There are three differences between the proposed method and previous methods. 1) Most previous methods only focus on self-attention, while we consider the cross-attention, which is proved to be more critical to translation quality (Voita et al., 2019; Tang et al., 2019; You et al., 2020; Ding et al., 2020b). 2) Previous methods usually multiply dot-product attention with a position-related bias or mask to model position. Our method additionally introduces a concentrated attention to compensate for dot-product attention, rather than simple bias. 3) Gaussian distribution is widely used in previous position modeling, while we use the more flexible GMM for complex cross-attention.

## 5 Experiments

We conducted experiments on three datasets and compare with the baseline and previous methods to evaluate the performance of the proposed method.

### 5.1 Datasets

Experiments were conducted on the following three datasets of different sizes.

**Nist Zh→En** 1.25M sentence pairs from LDC corpora<sup>1</sup>. We use MT02 as the validation set and MT03, MT04, MT05, MT06, MT08 as the test sets, each with 4 English references. Results are averaged on all test sets. We tokenize and lowercase English sentences with the Moses<sup>2</sup>, and segmented the Chinese sentences with Stanford Segmentor<sup>3</sup>. We apply BPE (Sennrich et al., 2016) with 30K merge operations on all texts.

**WMT14 En→De** 4.5M sentence pairs from WMT14<sup>4</sup> English-German task. We use news-test2013 (3000 sentence pairs) as the validation set and news-test2014 (3003 sentence pairs) as the test

<sup>1</sup>The corpora include LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

<sup>2</sup><https://www.statmt.org/moses/>

<sup>3</sup><https://nlp.stanford.edu/>

<sup>4</sup>[www.statmt.org/wmt14/](http://www.statmt.org/wmt14/)

set. We apply BPE with 32K merge operations, and the vocabulary is shared across languages.

**WMT17 Zh→En** 20M sentence pairs from WMT17<sup>5</sup> Chinese-English task, follow the processing of Hassan et al. (2018). We use devtest2017 (2002 sentence pairs) as the validation set and newstest2017 (2001 sentence pairs) as the test set. We apply BPE with 32K merge operations on all texts.

### 5.2 System

We conducted experiments on the following systems.

**Transformer** Baseline of our method. The architecture of Transformer-*Base/Big* was implemented strictly referring to Vaswani et al. (2017).

**Hard-Code Gaussian** Use a hard-code Gaussian distribution to replace dot-product attention in cross-attention (You et al., 2020). The hard-code Gaussian distribution is an artificially set Gaussian distribution with fixed mean and variance.

**Localness Gaussian** Our implementation of the modeling localness proposed by Yang et al. (2018). A learnable Gaussian bias is multiplied to model the local information of attention, especially in self-attention, and we apply it in cross-attention.

**Synthesis Network** Following (Graves, 2013), We modify the proposed Gaussian mixture attention with the synthesis network, which consists of  $K$  unnormalized Gaussian distributions, as shown in Table 1.

**Our Method** The proposed method. A Gaussian mixture attention is applied to cross-attention in Transformer. Refer to Sec.3 for specific details.

**Our Method+Norm.** Considering our method is approximate-normalized, we additionally propose strict-normalized Gaussian mixture attention, as a variant of our method. Since Gaussian mixture attention weights  $\beta_{ij}$  are all positive numbers, we directly use  $\beta_{ij} / \sum_{l=1}^n \beta_{lj}$  to normalize it, referring to ‘Our Method+Norm.’ in Table 1.

<sup>5</sup>[www.statmt.org/wmt17/](http://www.statmt.org/wmt17/)

Models	Nist Zh→En		WMT14 En→De		WMT17 Zh→En	
	BLEU	#Para.	BLEU	#Para.	BLEU	#Para.
<b>Transformer-Base</b>	44.02	79.7M	27.34	63.1M	24.03	83.6M
<b>Hard-Code Gaussian</b>	36.10	79.6M	25.03	63.0M	20.95	83.5M
<b>Localness Gaussian</b>	44.06	80.2M	27.41	63.6M	24.01	84.1M
<b>Synthesis Network</b>	43.34	79.8M	26.27	63.2M	23.69	83.7M
<b>Our Method+Norm.</b>	44.69↑	79.8M	27.50↑	63.2M	24.44↑	83.7M
<b>Our Method</b>	<b>45.39↑</b>	79.8M	<b>28.09↑</b>	63.2M	<b>24.44↑</b>	83.7M
<b>Transformer-Big</b>	44.20	247.5M	28.43	214.3M	24.46	255.2M
<b>Our Method (Big)</b>	<b>45.45↑</b>	247.6M	<b>29.02↑</b>	214.4M	<b>24.82↑</b>	255.3M

Table 2: BLEU score of our method and the existing NMT models on test sets. “#Para.”: the learnable parameter scale of the model (M=million). “↑”: the improvement is significant by contrast to baseline ( $\rho < 0.01$ ).

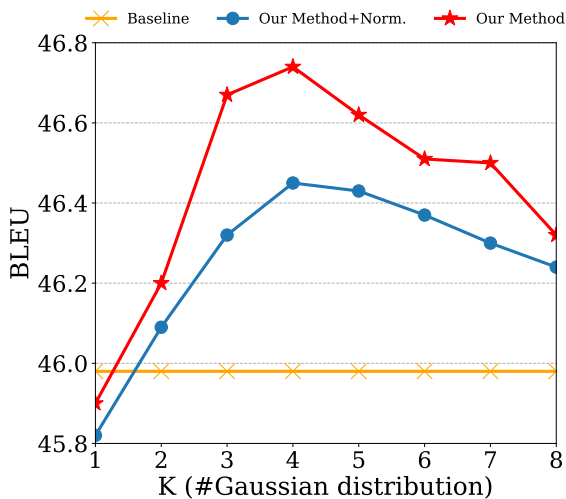


Figure 4: BLEU scores with different  $K$ .

The implementation is all adapted from Fairseq Library (Ott et al., 2019) with the same settings from Vaswani et al. (2017). SacreBLEU (Post, 2018) is applied to evaluate translation quality.

### 5.3 Effect of Hyperparameter $K$

Before the main experiment, as shown in Figure 4, we evaluate performance with various hyperparameter  $K$  on the Nist Zh→En validation set, where  $K$  represents the number of Gaussian distributions in the Gaussian mixture attention. When  $K = 1$ , since the cross-attention is not one-to-one, it is difficult for a single Gaussian distribution to fit the cross-attention. With the increase of  $K$ , the translation quality improves and performs best when  $K = 4$ . When  $K$  is large, too many Gaussian distributions in GMM-based attention will compli-

cate the model and maybe predict some inaccurate central words, resulting in a decrease in translation quality. Therefore, we set  $K = 4$  in the following experiments.

### 5.4 Main Results

Table 2 shows the results of our method compared with the baseline and previous methods. Our method achieves the best results on all three datasets, improving about 1.37 on Nist Zh→En, 0.75 on WMT14 En→De, and 0.41 on WMT17 Zh→En respectively, compared with Transformer-Base. Besides, compared to Transformer-Big, our method still brings significant improvements. Our method only increases 0.1% more parameters than Transformer-Base and achieve similar performance with Transformer-Big. The performance improvement of our method is not simply through increasing the model parameters, but to improve the cross-attention with the proposed Gaussian mixture attention.

‘Hard-Code Gaussian’ (You et al., 2020) and ‘Localness Gaussian’ (Yang et al., 2018) have been proved to be effective in self-attention but not obvious for cross-attention, since it’s difficult to fit complex cross-attention with their single Gaussian bias. The Gaussian mixture attention is more flexible and can fit arbitrarily complex distributions, especially to model multi-center attention distribution, which is more suitable for modeling cross-attention.

Considering normalization, ‘Synthesis Network’ is un-normalized, ‘Our Method+Norm.’ is strict-normalized, and ‘our method’ is approximate-normalized, where our proposed approximate

	BLEU	$\Delta$
<b>Our Method</b>	<b>28.09</b>	
– Share Mean	27.58	-0.51
– Share Variance	27.67	-0.42
– Share Weight	27.90	-0.19

Table 3: Performance when each Gaussian distribution shares the mean, variance, and weight, respectively.

normalization performs best. In practice, unnormalization tends to cause attention spikes, while strict normalization leads to unstable training.

## 6 Analysis

We conducted extensive analyses to understand the specific improvements of our method in attention entropy, alignment quality, phrase fluency, and long sentence translation. Unless otherwise specified, all the results are reported on WMT14 En→De test set with Transformer-Base.

### 6.1 Flexibility of Gaussian Mixture Attention

Compared with the single Gaussian distribution, GMM is more flexible on three aspects: mean, variance, and weight. To evaluate the improvements brought by these three aspects, we respectively share the mean, variance, and weight between each Gaussian distribution in Gaussian mixture attention, and report the results in Table 3.

The performance decreases most obviously when each Gaussian distribution sharing the same central word (mean). The major superiority of GMM over Gaussian distribution is that GMM contains multiple centers, which is more in line with cross-attention. The variance allows each central word to have different attention coverage, and the weight controls the contribution of each Gaussian distribution. The flexibility of these three aspects makes Gaussian mixture attention more suitable for cross-attention.

### 6.2 Effect of Gating Mechanism

Our method applies a gating mechanism to fuse Gaussian mixture attention and dot-product attention. We conduct the ablation study of directly averaging the two types of attention or only using one of them in Table 4. Our method surpasses only using a single type of attention or directly averaging the two types of attention, which shows that

	BLEU	$\Delta$
<b>Our Method</b>	<b>28.09</b>	
– Average Gating	27.61	-0.48
<b>Dot-product Attention</b>	27.34	-0.75
<b>Gaussian Mixture Attention</b>	26.32	-1.77

Table 4: Ablation study of directly averaging the two types of attention or only using one type of attention.

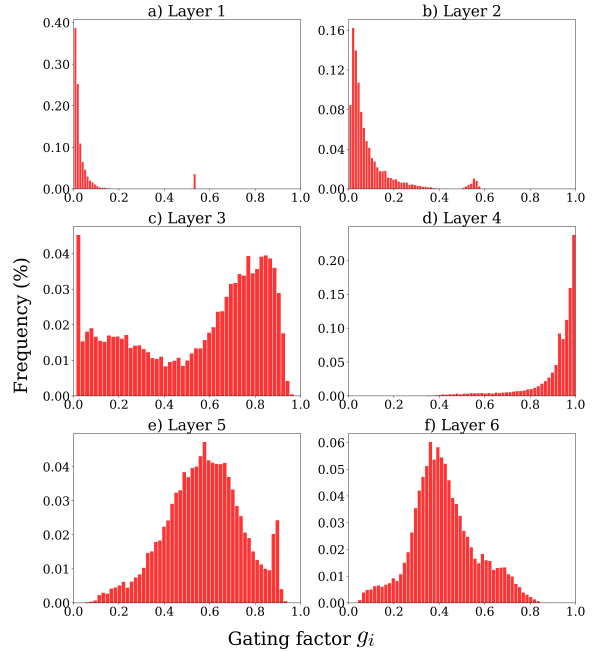


Figure 5: The distribution of gating factor  $g_i$  in the cross-attention of each layer. X-axis is  $g_i$ , which represents the weight of Gaussian mixture attention in the total attention, and Y-axis is the frequency with  $g_i$ .

the gating mechanism plays an important role and effectively fuses two types of attention.

To analyze the relationship between the two types of attention in detail, we calculate the distribution of gating factor ( $g_i$  in Eq.(13)) of each decoder layer, and show the result in Figure 5. To our surprise, our method makes the cross-attention in each decoder layer present a different division of labor, which confirms the conclusions of previous work (Li et al., 2019). Specifically, the bottom layers (L1, L2) in the decoder emphasize on dot-product attention and tend to capture global information; the middle layers (L3, L4) emphasize on Gaussian mixture attention, which captures local information around the central word; two types of attention in the top layer (L5, L6) are more balanced and jointly determine the final output. Previous work (Li et al., 2019) pointed out that the

		BLEU	$\Delta$
<b>Baseline</b>		27.34	
<b>Our method</b>	All 6 layers	28.09	+0.75
	Bottom 2 layers	27.43	+0.09
	Middle 2 layers	27.74	+0.40
	Top 2 layers	27.89	+0.55

Table 5: Result of using Gaussian mixture attention in different decoder layers.

		Entropy			
		All	Short (0,20]	Mid [21,40]	Long [41, $\infty$ ]
<b>Baseline</b>		2.67	2.40	2.80	2.94
<b>Ours</b>	GMA	0.57	0.59	0.55	0.50
	DP	2.68	2.42	2.80	2.90
	Total	1.86	1.78	1.90	1.90

Table 6: Entropy of attention distribution on varying source sentence length. ‘GMA’: Gaussian mixture attention. ‘DP’: Dot-product attention. ‘Total’: Fusion of two types of attention.

cross-attention of each layer is different, where the lower layer tends to capture the sentence information, while the upper layer tends to capture the alignment and specific word information since it is closer to the output. Our method also confirms this point, where Gaussian mixture attention effectively model concentrated attention so that it occupies a larger proportion in higher layers. With the gating mechanism, our method successfully fuses two types of attention and learns the division of labor between different layers.

Based on this, we tried to only apply our method to a part of decoder layers to verify the effect of our method on different layers, and the results are reported in Table 5. When Gaussian mixture attention is only used in the top or middle 2 layers, the translation quality can be significantly improved without requiring many additional calculations compared with *Transformer-Base*.

### 6.3 Entropy of Attention Distribution

We use Gaussian mixture attention to model concentrated attention to make up for the dispersion of dot-product attention, especially on the long source. Entropy is often used to measure the dispersion of distribution, where the higher entropy means that the distribution is more dispersed (He et al., 2020).

	AER	P.	R.
<b>Transformer-Base</b>	53.79	47.75	47.09
<b>Our method</b>	<b>48.21</b>	<b>53.94</b>	<b>52.83</b>
<b>Transformer-Big</b>	46.50	54.46	56.15
<b>Our method (Big)</b>	<b>43.03</b>	<b>58.86</b>	<b>60.62</b>

Table 7: Alignment quality of our method and baseline.

We report the entropy of the attention distribution in our method on varying source sentence length in Table 6. The entropy of dot-product attention increase with the length of the sentence, showing that dot-product attention is easy to become dispersed as the source length increases. However, the entropy of concentrated attention modeled by Gaussian mixture attention always remains at a low level, since it’s unaffected by the source length. Overall, with the proposed Gaussian mixture attention, our method has lower entropy than baseline, indicating that our method focuses more on some important words, which proves to be beneficial to translation (He et al., 2020; Zhang et al., 2019).

### 6.4 Alignment Quality

The Gaussian mixture attention we proposed explicitly models the concentrated attention, so it is potential to help cross-attention achieve more accurate alignment between the target and the source. To explore this conjecture, we evaluate the alignment accuracy of our method on RWTH En→De alignment dataset<sup>6</sup> (Liu et al., 2016; Ghader and Monz, 2017; Tang et al., 2019).

Following Luong et al. (2015) and Kuang et al. (2018), we force the models to produce the reference target words during inference to get the attention between source and target. We average the attention weights across all heads from the penultimate layer (Li et al., 2019; Ding et al., 2020a), where the source token with the highest attention weight is viewed as the alignment of the current target token. The alignment error rate (AER) (Och and Ney, 2003), precision (P.), and recall (R.) of our method are reported in Table 7.

Our method achieves better alignment accuracy than baseline, improving 5.58 on *Transformer-Base* and 3.47 on *Transformer-Big*, which shows that modeling concentrated attention indeed improves the alignment quality of cross-attention.

<sup>6</sup><https://www-i6.informatik.rwth-aachen.de/goldAlignment/>



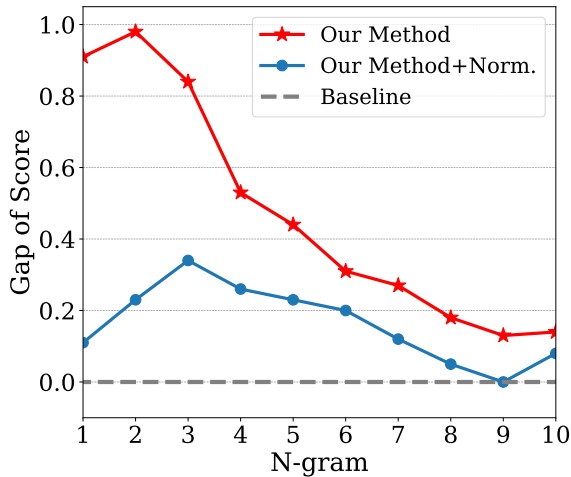


Figure 6: Gap of N-gram score between our method and baseline with respect to various N, to highlight the difference in improvement of different n-grams.

### 6.5 N-gram Accuracy

Our method uses Gaussian mixture attention to model concentrated attention, which intuitively should be able to enhance the ability to capture neighboring structures, thereby obtaining more fluent translation. To evaluate the quality of phrase translation, we calculate the improvement of our method on various N-grams in Figure 6. We set *Transformer-Base* as the baseline, and report the gap of score between our method and baseline on each N-gram score. Our method is superior to the baseline in all N-grams, especially on 2-gram and 3-gram, which shows that our method effectively captures the nearby phrase structure. In the concentrated attention modeled by Gaussian mixture attention, the attention to the surrounding words increases along with the central word, resulting in better phrase translation.

### 6.6 Analysis on Sentence Length

To analyze the improvement of our method on sentences with different lengths, we group the sentences into 6 sets according to the source length (Bahdanau et al., 2014; Tu et al., 2016), and report the BLEU scores on each set in Figure 7.

Compared with Baseline, our method has a more significant improvement in long sentences, with +1.36 BLEU on (30, 40], +1.06 BLEU on (40, 50], and +4.14 BLEU on (50, +∞]. Our method significantly improves the long sentence translation by modeling concentrated cross-attention. When the source sentence is very long, dot-product attention fairly pays attention to every source word

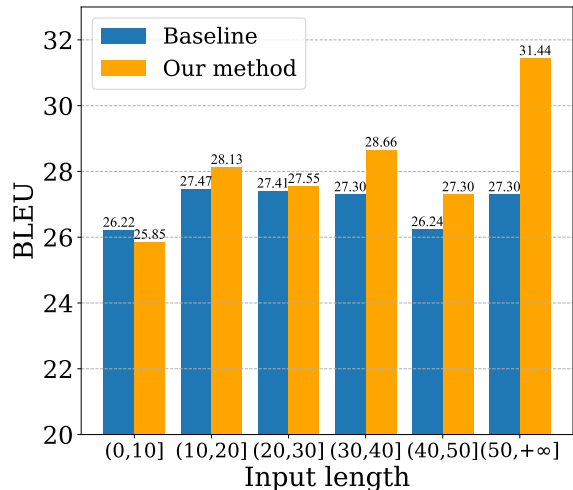


Figure 7: BLEU scores of sentence with various length.

and normalizes it, causing cross-attention to become dispersed, which proved to be unfavorable for translation in previous work (Zhang et al., 2019; He et al., 2020; Tang et al., 2019). In contrast, regardless of the length of the source, Gaussian mixture attention concentrates surround some central words, effectively avoiding the attention dispersion. Therefore, our method effectively improves the translation quality of long sentences by modeling the concentrated attention. Besides, our method drops slightly when the sentence length is small. Since we set  $K = 4$  for the source sentences of different length, when the sentence is very short, it is difficult to accurately find 4 corresponding central words (mean) for each target word.

## 7 Conclusion

Inspired by linguistics, we decompose the cross-attention into distributed attention and concentrated attention. To model the concentrated attention, we apply GMM to construct the Gaussian mixture attention, which effectively resolves the weakness of dot-product attention. Experiments show that the proposed method outperforms the strong baseline on three datasets. Further analyses show the specific advantages of the proposed method in attention distribution, alignment quality, N-gram accuracy, and long sentence translation.

### Acknowledgements

We thank all the anonymous reviewers for their insightful and valuable comments. This work was supported by National Key R&D Program of China (NO. 2017YFE0192900).

## References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. [Neural machine translation by jointly learning to align and translate](#). Cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation.
- E. Battenberg, R. J. Skerry-Ryan, S. Mariooryad, D. Stanton, D. Kao, M. Shannon, and T. Bagby. 2020. [Location-relative attention mechanisms for robust long-form speech synthesis](#). In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6194–6198.
- John Brand and Aaron P. Johnson. 2018. [The effects of distributed and focused attention on rapid scene categorization](#). *Visual Cognition*, 26(6):450–462.
- Liang Ding, Longyue Wang, and Dacheng Tao. 2020a. [Self-attention with cross-lingual position representation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1679–1685, Online. Association for Computational Linguistics.
- Liang Ding, Longyue Wang, Di Wu, Dacheng Tao, and Zhaopeng Tu. 2020b. [Context-aware cross-attention for non-autoregressive translation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4396–4402, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Hamidreza Ghader and Christof Monz. 2017. [What does attention in neural machine translation pay attention to?](#) In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 30–39, Taipei, Taiwan. Asian Federation of Natural Language Processing.
- Alex Graves. 2013. [Generating sequences with recurrent neural networks](#). *CoRR*, abs/1308.0850.
- Hany Hassan, Anthony Aue, C. Chen, Vishal Chowdhary, J. Clark, C. Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, W. Lewis, M. Li, Shujie Liu, T. Liu, Renqian Luo, Arul Menezes, Tao Qin, F. Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and M. Zhou. 2018. [Achieving human parity on automatic chinese to english news translation](#). *ArXiv*, abs/1803.05567.
- Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. 2020. [RealFormer: Transformer Likes Residual Attention](#). *arXiv e-prints*, page arXiv:2012.11747.
- Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. 2020. [Realformer: Transformer likes residual attention](#).
- Jinbae Im and Sungzoon Cho. 2017. [Distance-based self-attention network for natural language inference](#).
- Minami Ito, Gerald Westheimer, and Charles D Gilbert. 1998. [Attention and perceptual learning modulate contextual influences on visual perception](#). *Neuron*, 20(6):1191 – 1197.
- Beck Jacob and Ambler Bruce. 1973. [The effects of concentrated and distributed attention on peripheral acuity](#). *Perception & Psychophysics*, 14(2).
- Shaohui Kuang, Junhui Li, António Branco, Weihua Luo, and Deyi Xiong. 2018. [Attention focusing for neural machine translation by bridging source and target embeddings](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1767–1776, Melbourne, Australia. Association for Computational Linguistics.
- Xintong Li, Guanlin Li, Lemao Liu, Max Meng, and Shuming Shi. 2019. [On the word alignment from neural machine translation](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1293–1303, Florence, Italy. Association for Computational Linguistics.
- Lemao Liu, Masao Utiyama, Andrew Finch, and Ei-ichiro Sumita. 2016. [Neural machine translation with supervised attention](#). In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102, Osaka, Japan. The COLING 2016 Organizing Committee.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. [Effective approaches to attention-based neural machine translation](#). In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.
- Franz Josef Och and Hermann Ney. 2003. [A systematic comparison of various statistical alignment models](#). *Computational Linguistics*, 29(1):19–51.
- Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. [fairseq: A fast, extensible toolkit for sequence modeling](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Friedrich Pukelsheim. 1994. [The three sigma rule](#). *The American Statistician*, 48(2):88–91.

- Carl Edward Rasmussen. 1999. [The infinite gaussian mixture model](#). In *NIPS*, pages 554–560. The MIT Press.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. [Neural machine translation of rare words with subword units](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.
- Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. [Self-attention with relative position representations](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics.
- Kaitao Song, Xu Tan, Furong Peng, and Jianfeng Lu. 2018. [Hybrid self-attention network for machine translation](#).
- Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. 2018. [Self-attentional acoustic models](#). pages 3723–3727.
- Gongbo Tang, Rico Sennrich, and Joakim Nivre. 2019. [Understanding neural machine translation by simplification: The case of encoder-free models](#). In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 1186–1193, Varna, Bulgaria. INCOMA Ltd.
- Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. [Modeling coverage for neural machine translation](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. Association for Computational Linguistics.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. [Attention is all you need](#). In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.
- Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. [Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.
- Xing Wang, Zhaopeng Tu, Longyue Wang, and Shuming Shi. 2019. [Self-attention with structural position representations](#). 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)*, pages 1403–1409, Hong Kong, China. Association for Computational Linguistics.
- Mingzhou Xu, Derek F. Wong, Baosong Yang, Yue Zhang, and Lidia S. Chao. 2019. [Leveraging local and global patterns for self-attention networks](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3069–3075, Florence, Italy. Association for Computational Linguistics.
- Baosong Yang, Zhaopeng Tu, Derek F. Wong, Fandong Meng, Lidia S. Chao, and Tong Zhang. 2018. [Modeling localness for self-attention networks](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4449–4458, Brussels, Belgium. Association for Computational Linguistics.
- Weiyou You, Simeng Sun, and Mohit Iyyer. 2020. [Hard-coded gaussian attention for neural machine translation](#). pages 7689–7700.
- J. Zhang, Y. Zhao, H. Li, and C. Zong. 2019. [Attention with sparsity regularization for neural machine translation and summarization](#). *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(3):507–518.