

Attention Calibration for Transformer in NMT

ACL 2021

2022.7.11

Abstract

- 提出的问题: the attention mechanisms' capability for discovering decisive inputs
- 提出的模型: mask perturbation model & attention calibration network
- 主要结果:
 - 模型好
 - 注意力权重的分布规律
 - 什么时候需要去calibrate?

Abstract

Attention mechanisms have achieved substantial improvements in neural machine translation by dynamically selecting relevant inputs for different predictions. However, recent studies have questioned the attention mechanisms' capability for discovering decisive inputs. In this paper, we propose to calibrate the attention weights by introducing a mask perturbation model that automatically evaluates each input's contribution to the model outputs. We increase the attention weights assigned to the indispensable tokens, whose removal leads to a dramatic performance decrease. The extensive experiments on the Transformer-based translation have demonstrated the effectiveness of our model. We further find that the calibrated attention weights are more uniform at lower layers to collect multiple information while more concentrated on the specific inputs at higher layers. Detailed analyses also show a great need for calibration in the attention weights with high entropy where the model is unconfident about its decision¹.

Conclusion

- 模型好
- mask perturbation model 可以找到input中最重要的信息
- 三种方法做calibration
- 注意力权重分布规律：
 - Low layers: uniform
 - High layers: concentrated
- Attention wights with high entropy

7 Conclusion

In this paper, we present a mask perturbation model to automatically discover the decisive inputs for the model prediction. We propose three methods to calibrate the attention mechanism by focusing on the discovered vital inputs. Extensive experimental results show that our approaches obtain significant improvements over the state-of-the-art system. Analytical results indicate that our proposed methods make the low layer's attention weights more dispersed to grasp multiple information. In contrast, high-layer attention weights become more

focused on specific essential inputs. We further find a greater need for calibration in the original attention weights with high entropy. Our work provides insights on future work about learning more useful information via attention mechanisms in other attention-based frameworks.

Introduction

想法来源：

1. On the one hand, erasing the representations accorded high attention weights do not necessarily lead to a performance decrease (*Serrano and Smith, 2019*)
2. On the other hand, *Jain and Wallace (2019)* state that attention weights are inconsistent with other feature importance metrics in text classification tasks.

Introduction

mask perturbation model

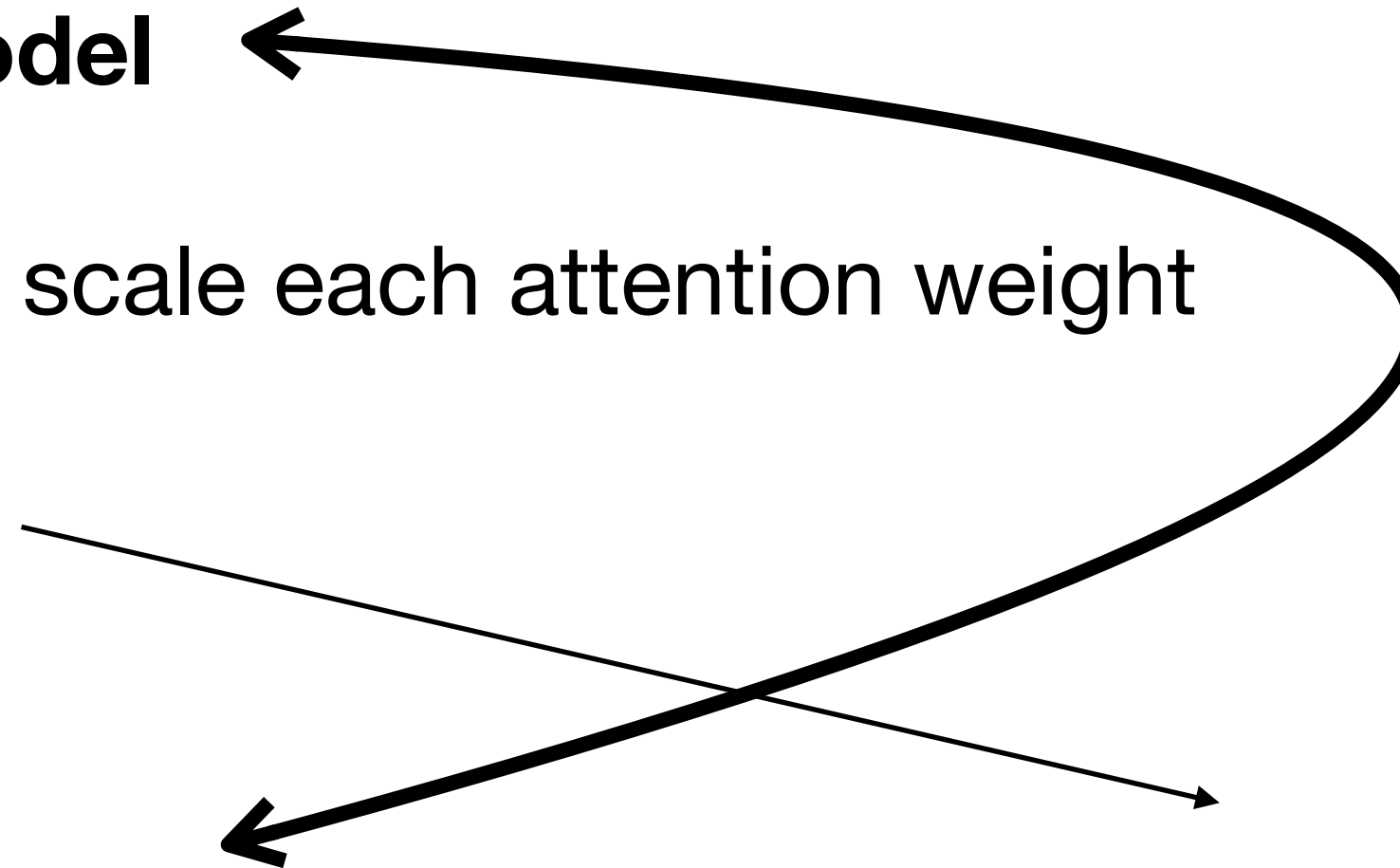
- A **learnable** mask to scale each attention weight
- “A deletion game”

Jointly trained

How to calibrate ?

- Fixed weighted sum
- Annealing learning
- Gating mechanism

The smallest perturbation extents that
cause the significant quality degradation



Innovation

Can attention be improved ?

- Lexical probabilities
- Word alignment
- Human rationales
- Sparsity regularization

we never introduce any external knowledge but highlight the inputs whose removal would significantly decrease Transformer's performance.

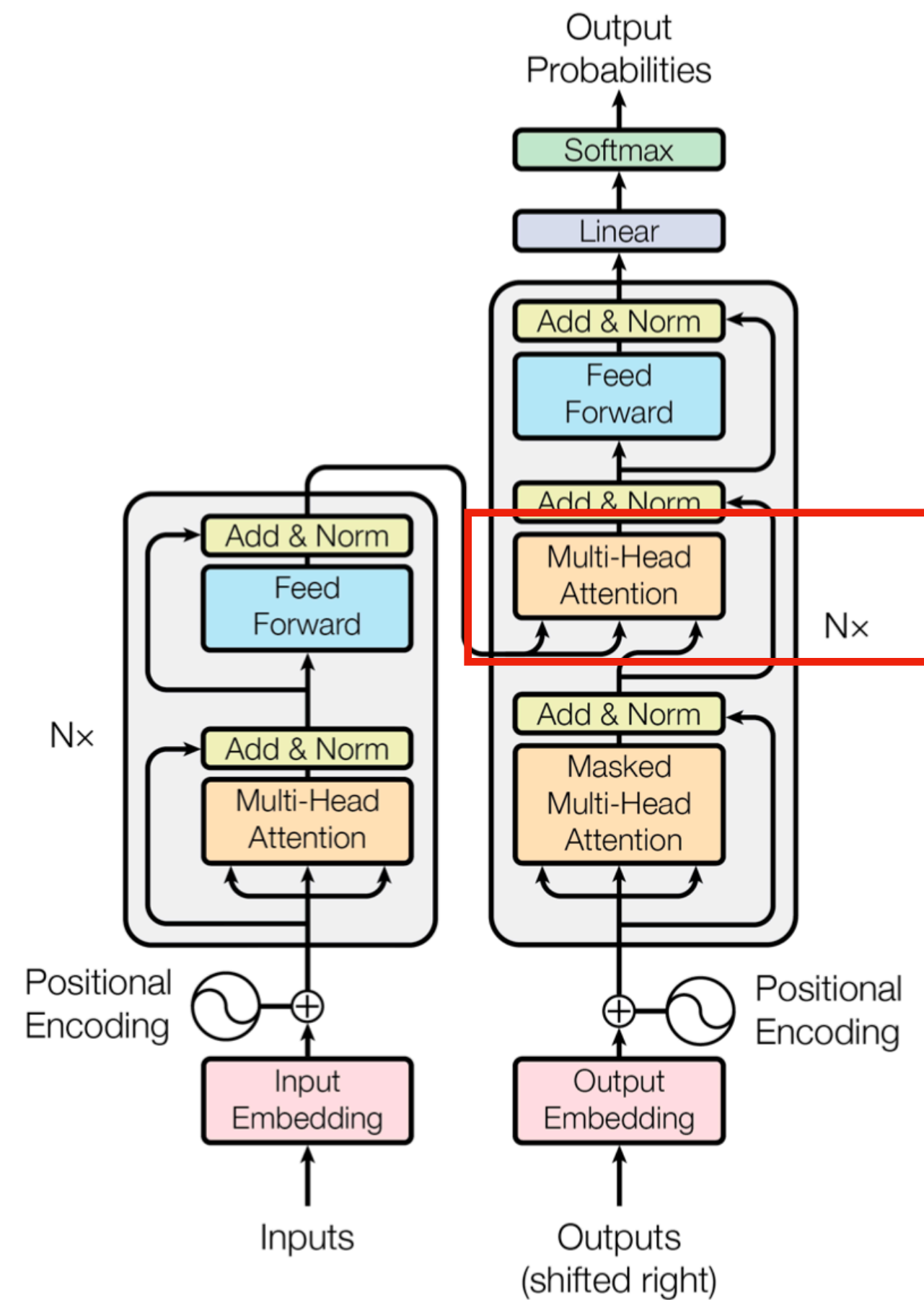
Another work line aims to make attention better indicative of the inputs' importance

本工作不仅对输入做了分析，同时也利用该分析结果使得性能提升。

Background

$$\text{Attn}(\mathbf{q}_t, \mathbf{K}, \mathbf{V}) = \alpha_t \mathbf{V}$$

$$\alpha_t = \text{softmax}\left(\frac{\mathbf{q}_t \mathbf{K}^T}{\sqrt{d_k}}\right)$$



Multi-head attention between encoder and decoder enables each prediction to attend overall inputs from different representation subspaces jointly.

Figure 1: The Transformer - model architecture.

Background

Our analysis examines the correlation with **attention weights** and **feature importance metrics** in NMT to test if the attention mechanisms focus on the decisive inputs.

Gradient-based methods

$$\tau_{it} = \left| \nabla_{h_i} p(y_t \mid \mathbf{x}_{1:n}) \right|$$

measure the importance of each contextual representation h_i for model output y_t

$$P(y_t \mid \mathbf{x}) = \frac{e^{\mathbf{x}^\top \mathbf{w}_j}}{\sum_{k=1}^K e^{\mathbf{x}^\top \mathbf{w}_k}}$$

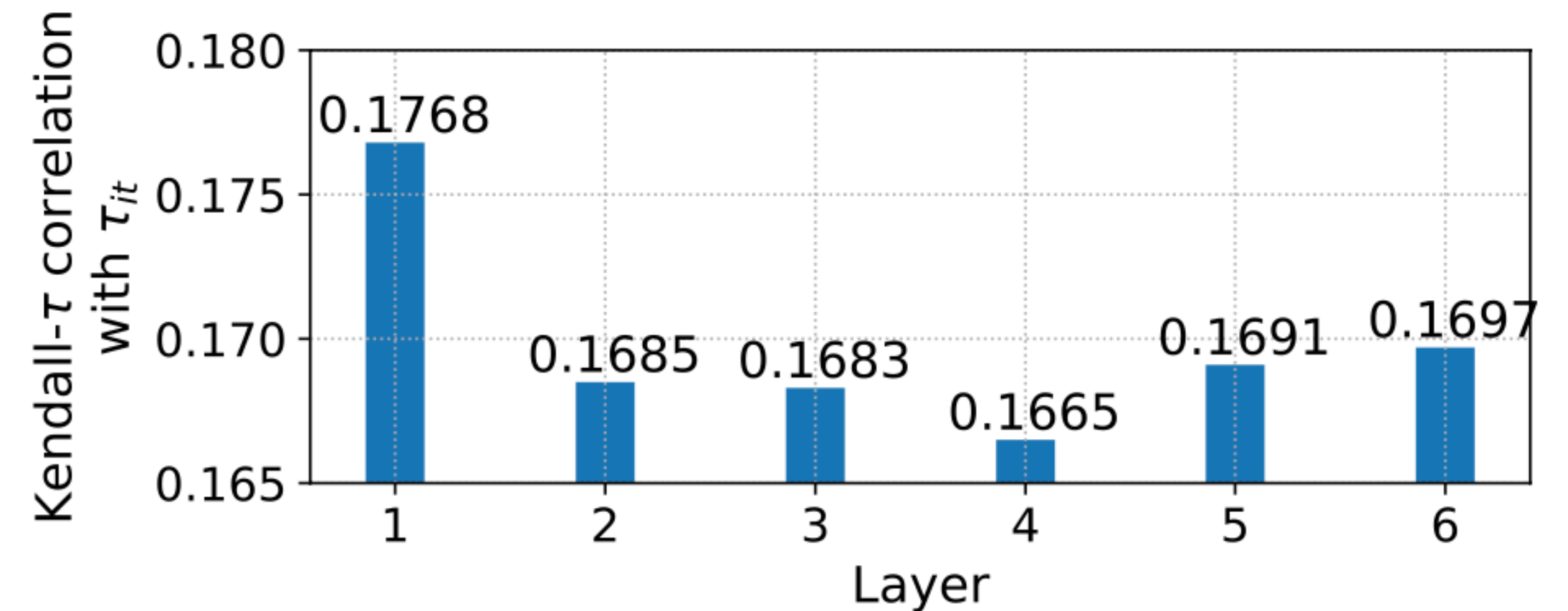
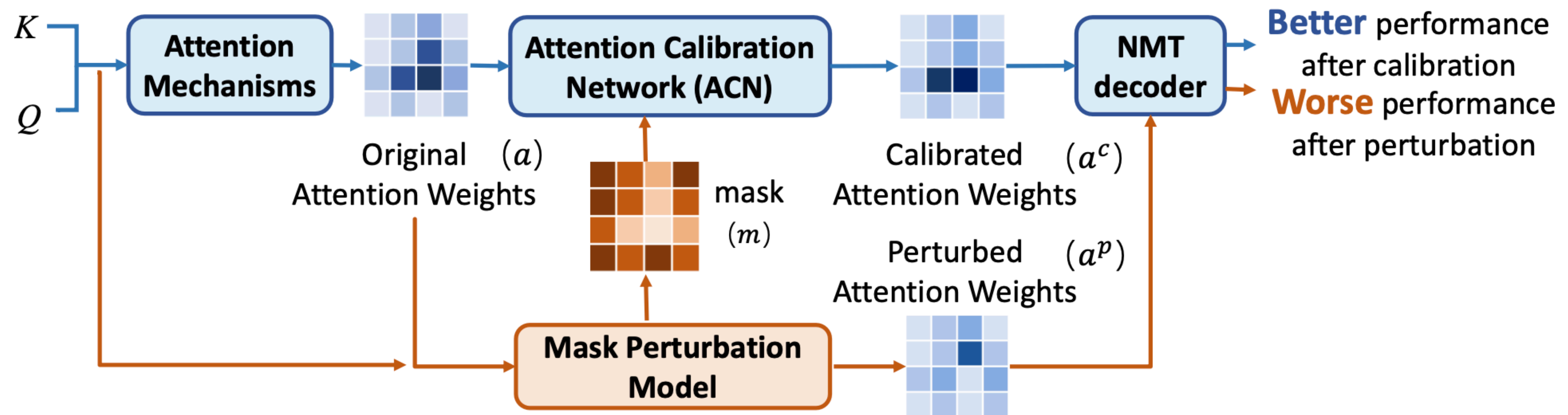


Figure 2: The mean **Kendall- τ** correlation between attention weights (a) and gradient importance metrics (τ_{it}) on Zh \Rightarrow En translation.

Our Method

Attention Calibration Network: to correct the original attention weights, highlighting the decisive inputs based on what inputs are perturbed by the mask perturbation model



Mask Perturbation Model: By doing this, we can automatically detect what inputs decide the model outputs.

Our Method

Src: 远郊连日大雪多人死亡交通中断

Ref: days of heavy snow in countryside left many deaths and **transportation disrupted**

Base: heavy snow in countryside caused many deaths

Ours: heavy snow *in countryside* has caused many *deaths and* traffic interruption

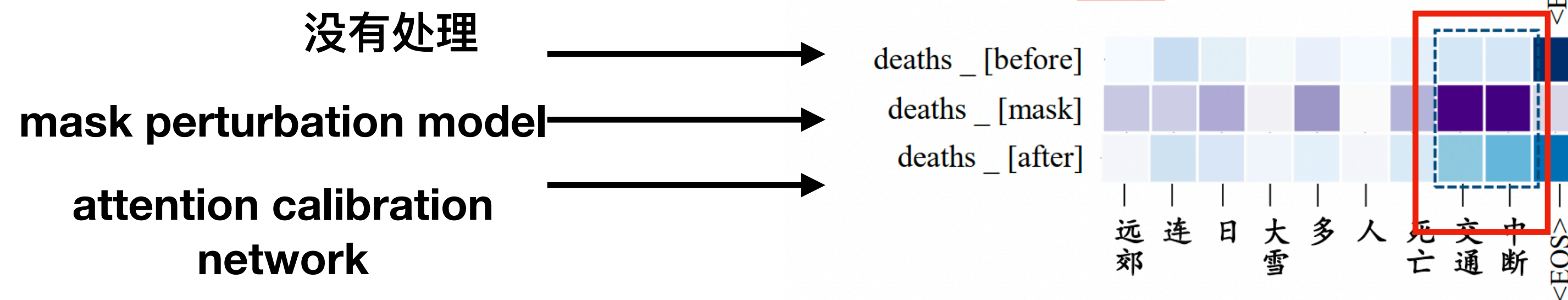


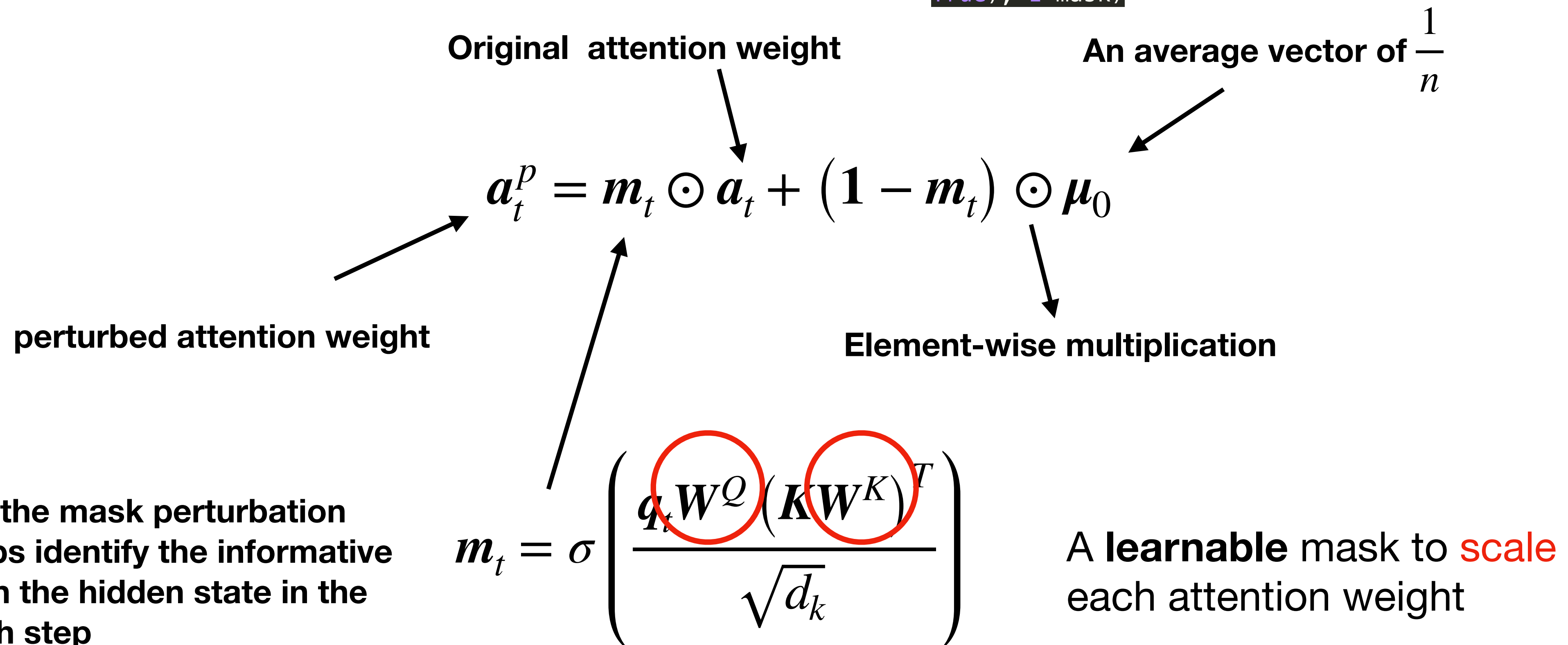
Figure 1: Examples of the attention weights before and after calibration. “in _” denotes the timestep after the prediction “in”. The dashed boxes indicate the inputs which should receive more attention measured by our mask perturbation model.

Our Method

Mask Perturbation Model

apply a mask to scale each input's attention weight, which simulates the process of perturbation.

```
torch.mul(torch.mean(attn_weights_float, -1, True), 1-mask)
```



Our Method

Mask Perturbation Model

【Cross-entropy loss】 when using perturbed attention weight
Harm the translation quality

$\mathcal{L}(\theta^m) = -\mathcal{L}_{\text{NMT}}(\mathbf{a}_t^p, \theta) + \alpha \mathcal{L}_c(\theta^m)$

A penalty to encourage most of mask to be turned off

The parameters of the original transformer

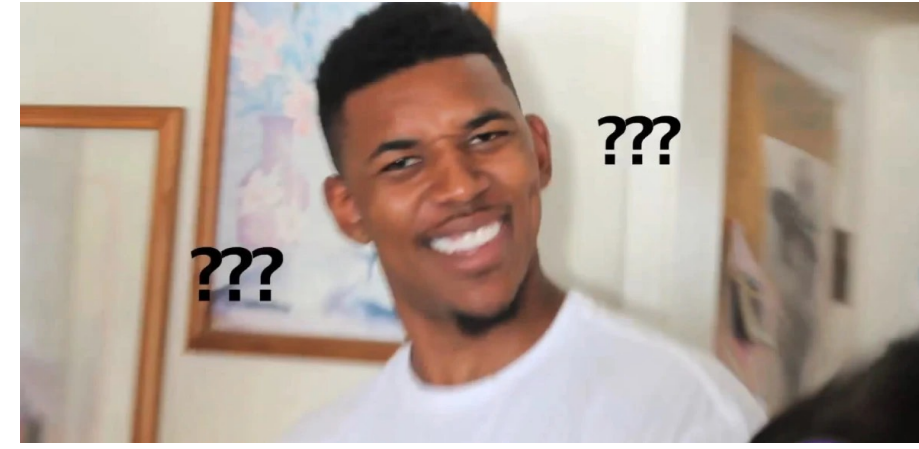
A large α forces the model to turn off most masks

$$\theta^m = \{W^Q, W^K\}$$

$$\mathcal{L}_c(\theta^m) = \|\mathbf{1} - \mathbf{m}_t\|_2$$

Our Method

Attention Calibration Network



Formally, the **calibrated attention weight** a_t^c can be designed as:

$$\text{calibrated attention weight} \quad a_t^c = a_t \odot e^{1-m_t} \quad (7)$$

$$\text{Fixed Weighted Sum} \quad a_t^{comb} = \text{softmax} \left(a_t + \lambda * a_t^c \right) \quad \text{Hyperparameter}$$

We increase the attention weights of key inputs which suffer large perturbation extents. The attention weights of other less-informative inputs are correspondingly decreased. We design three methods to incorporate a_t^c into the original one a_t to obtain **combined attention weights** a_t^{comb} :

$$\text{Annealing Learning} \quad a_t^{comb} = \gamma(s) * a_t + (1 - \gamma(s)) * a_t^c \quad \gamma(s) = e^{-s/10^5}$$

$$\text{Gated Mechanism} \quad a_t^{comb} = g_t * a_t + (1 - g_t) * a_t^c \quad g_t = \sigma(q_t W^g + b^g) \quad \text{Trainable parameters}$$


$$\mathcal{L}_{\text{NMT}}(\theta) = - \sum_{t=1}^m \log p(y_t | y_{<t}, x; a_t^{comb}, \theta)$$

Experiments

Datasets

Tokenize the
corpora using a
script from Moses

Byte pair
Encoding

| Source | Lang. | Train | Dev. | Test | Vocab. |
|--------------------|--|-------|------|--|--------|
| LDC ¹ | Zh \Rightarrow En | 2.09M | 878 | 4789 | 32k |
| WMT14 ² | En \Rightarrow De | 4.54M | 3000 | 3003  | 37k |
| WMT17 ³ | En \Rightarrow Lv Lv \Rightarrow En | 4.46M | 2003 | 2001 | 37k |
| | En \Rightarrow Fi Fi \Rightarrow En | 2.63M | 3000 | 3002 | 32k |
| WMT16 ⁴ | En \Rightarrow Ro Ro \Rightarrow En | 0.61M | 1999 | 1999 | 32k |

Experiments

Main results

| Model | | TEST |
|----------------------------------|--------|--------------|
| GNMT (Wu et al., 2016)‡ | | 24.61 |
| Conv (Gehring et al., 2017)‡ | | 25.16 |
| AttIsAll (Vaswani et al., 2017)‡ | | 27.3 |
| (Feng et al., 2020)‡ | | 27.55 |
| (Weng et al., 2020)‡ | | 27.7 |
| Our Implemented Baseline | | 27.37 |
| Ours | Fixed | 27.38 |
| | Anneal | 28.1* |
| | Gate | 27.75 |

Table 2: The comparison of our model, Transformer baselines and related work on the WMT14 En⇒De using case-sensitive BLEU. Results with [‡] mark are taken from the corresponding papers. “*” indicates the gains are statistically significant than baselines with $p<0.05$.

| Model | | DEV | MT03 | MT04 | MT05 | MT06 | AVE |
|----------|--------|---------------|---------------|---------------|---------------|---------------|---------------|
| Baseline | | 48.56 | 49.58 | 48.58 | 49.95 | 47.22 | 48.24 |
| Ours | Fixed | 48.42 | 49.41 | 48.56 | 50.32 | 47.89 | 48.44 |
| | Anneal | 48.22 | 49.73 | 48.85 | 50.97* | 47.49 | 48.74 |
| | Gate | 49.52* | 50.42* | 49.16* | 50.78* | 47.98* | 49.00* |

Table 3: Evaluation of translation quality for Zh⇒En Translation using case-insensitive BLEU score. “*” indicates the gains are statistically significant than baselines with $p<0.05$.

| Model | | En⇒Lv | Lv⇒En | En⇒Fi | Fi⇒En | En⇒Ro | Ro⇒En |
|----------|--------|---------------|---------------|---------------|---------------|---------------|---------------|
| Baseline | | 16.26 | 17.76 | 22.01 | 26.07 | 22.56 | 27.53 |
| Ours | Fixed | 16.54 | 18.45* | 22.42 | 26.2 | 23.1 | 28.02 |
| | Anneal | 16.35 | 18.12 | 22.4 | 26.39 | 23.27* | 28.2* |
| | Gate | 16.83* | 18.71* | 22.55* | 26.67* | 24.00* | 28.48* |

Table 4: Evaluation of translation quality for WMT17 En⇔Fi, WMT17 En⇔Lv and WMT16 En⇔Ro using case-insensitive BLEU score. “*” indicates the gains are statistically significant than baselines with $p<0.05$.

Small-scale

Experiments

Main results

Effect of Hyperparameter

$$\mathcal{L}(\theta^m) = -\mathcal{L}_{\text{NMT}}(a_t^p, \theta) + \alpha \mathcal{L}_c(\theta^m)$$

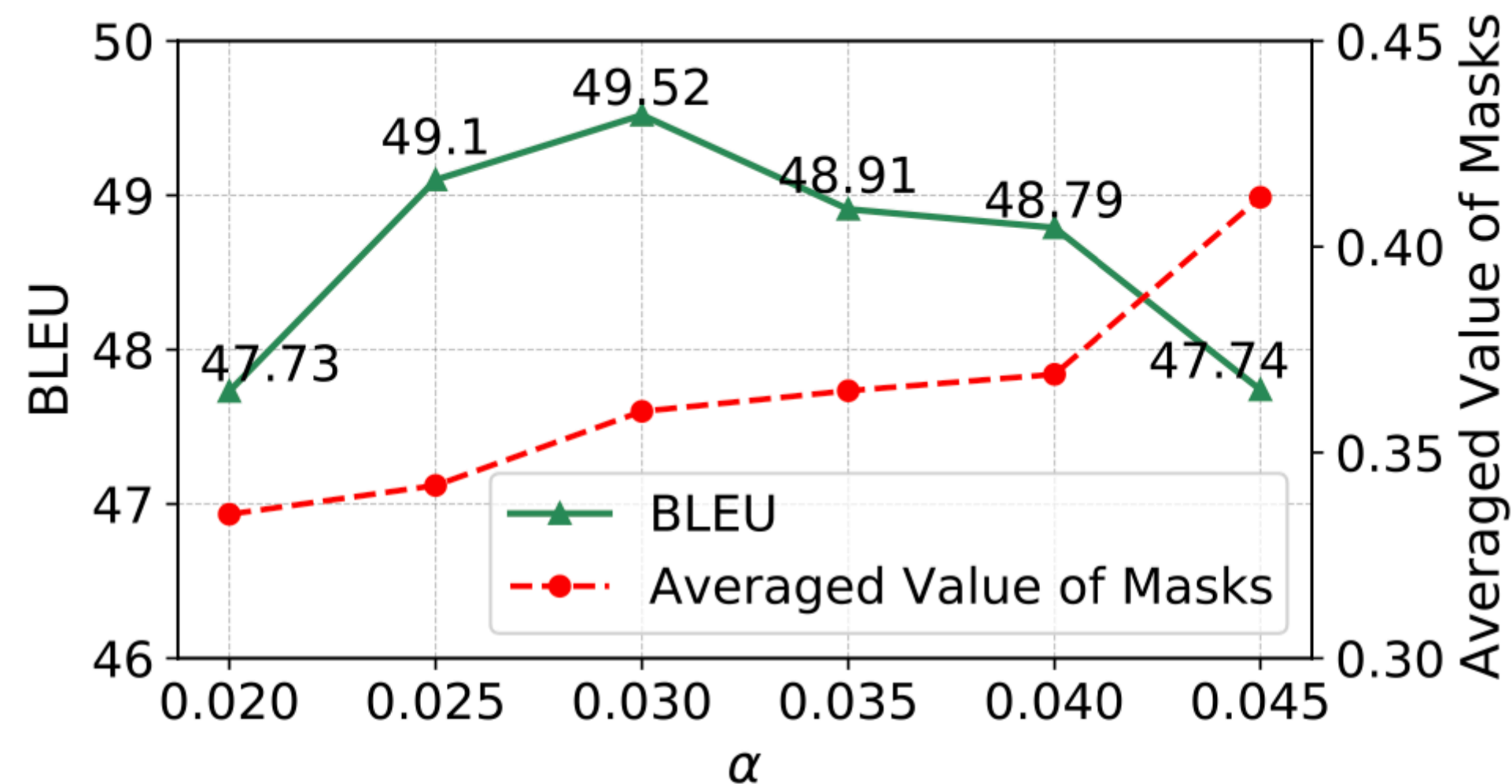


Figure 4: Experimental results on the validation set and the averaged value of generated masks with respect to different hyperparameter α on Zh \Rightarrow En translation task (Gate Mechanism).

Mask perturbation model is effective

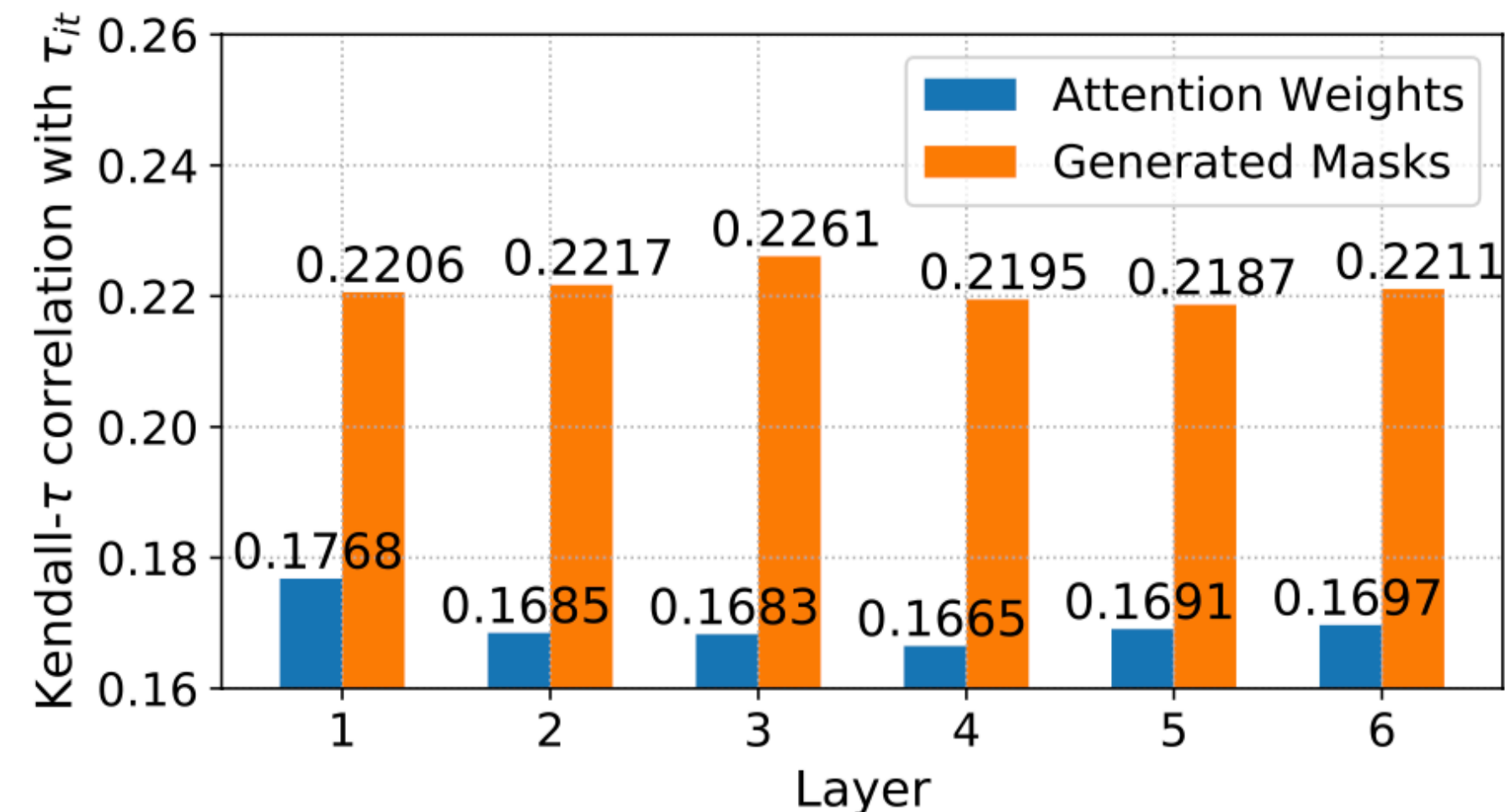


Figure 5: The mean **Kendall- τ** correlation between attention weights (a), the masks (m) generated by our mask perturbation model and gradient importance measures (τ_{it}) on Zh \Rightarrow En.

Analysis

What attention weights need to calibrate ?

A high JSD means the calibrated attention weights are distant from the original one

$$\text{JSD} (a_1, a_2) = \frac{1}{2} \text{KL} [a_1 \| \bar{a}] + \frac{1}{2} \text{KL} [a_2 \| \bar{a}]$$

$$\bar{a} = \frac{a_1 + a_2}{2}$$

$$D_{KL}(p \| q) = \sum_{j=1}^n p(x_j) \ln \frac{p(x_j)}{q(x_j)}$$

test whether the calibrated attention weights become more uniform or focused

$$\Delta Ent(a_1, a_2) = ent(a_1) - ent(a_2)$$

$$ent(a) = - \sum_{i=1}^m a_i \log a_i$$

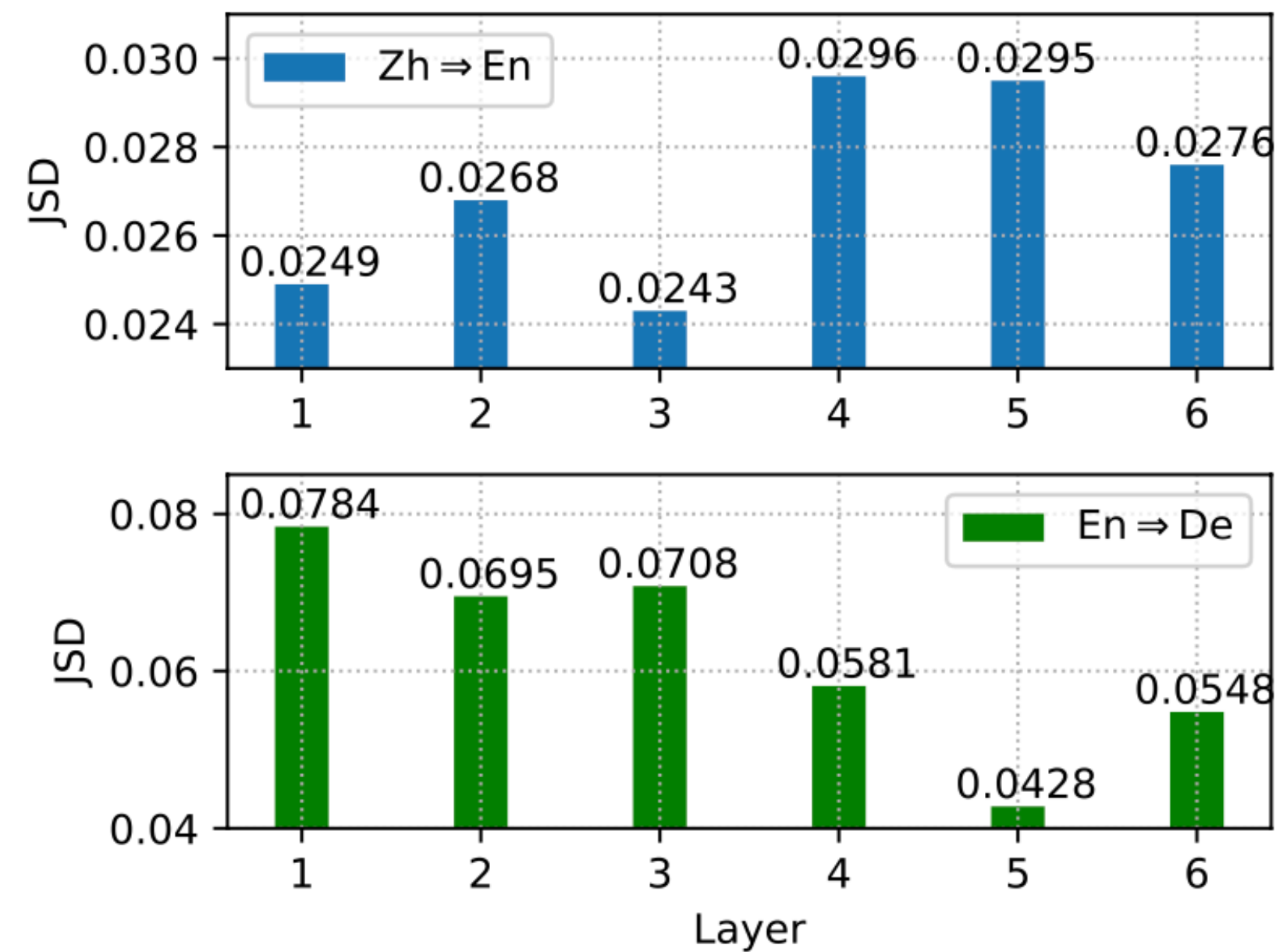
A metric to describe the uncertainty of the distribution

Analysis

What attention weights need to calibrate ?

More focused contributions of inputs suggest that the model is more confident about the choice of important tokens (Voita et al., 2020).

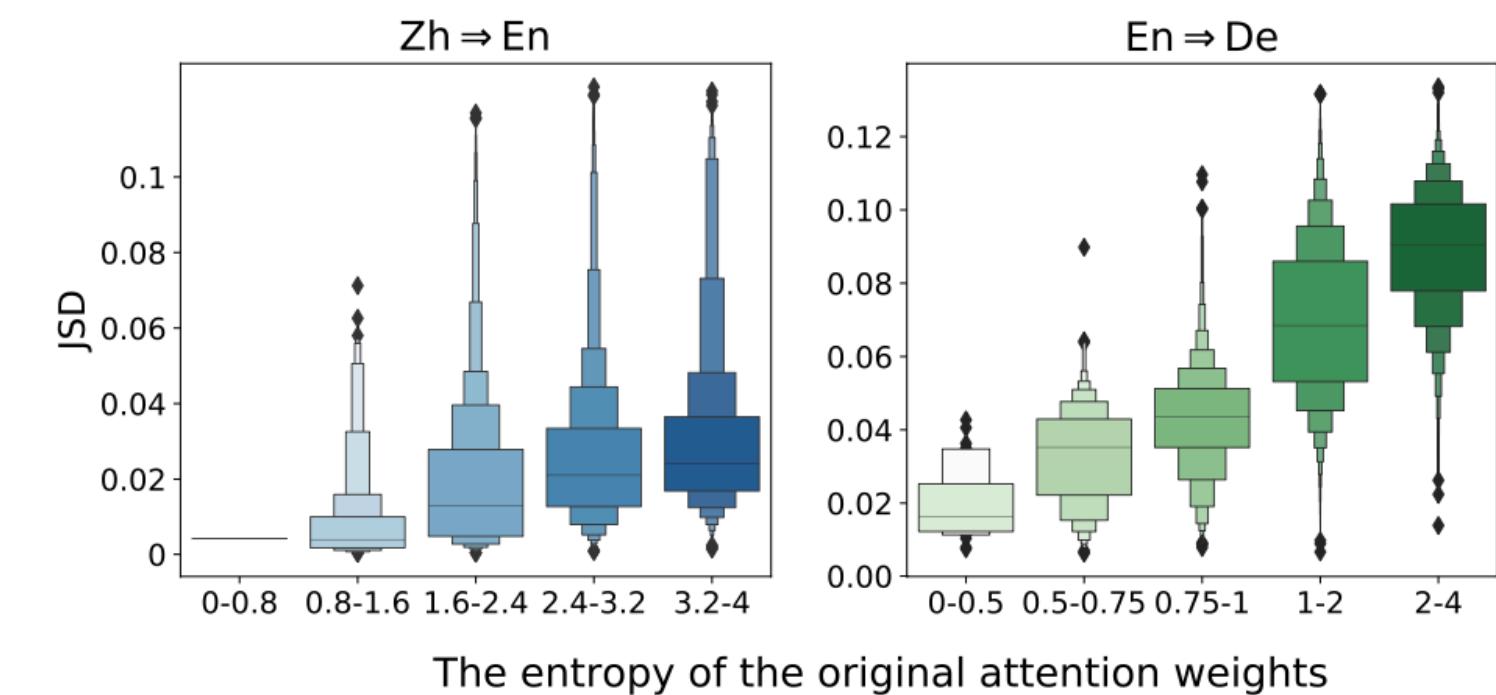
High or low layers ?



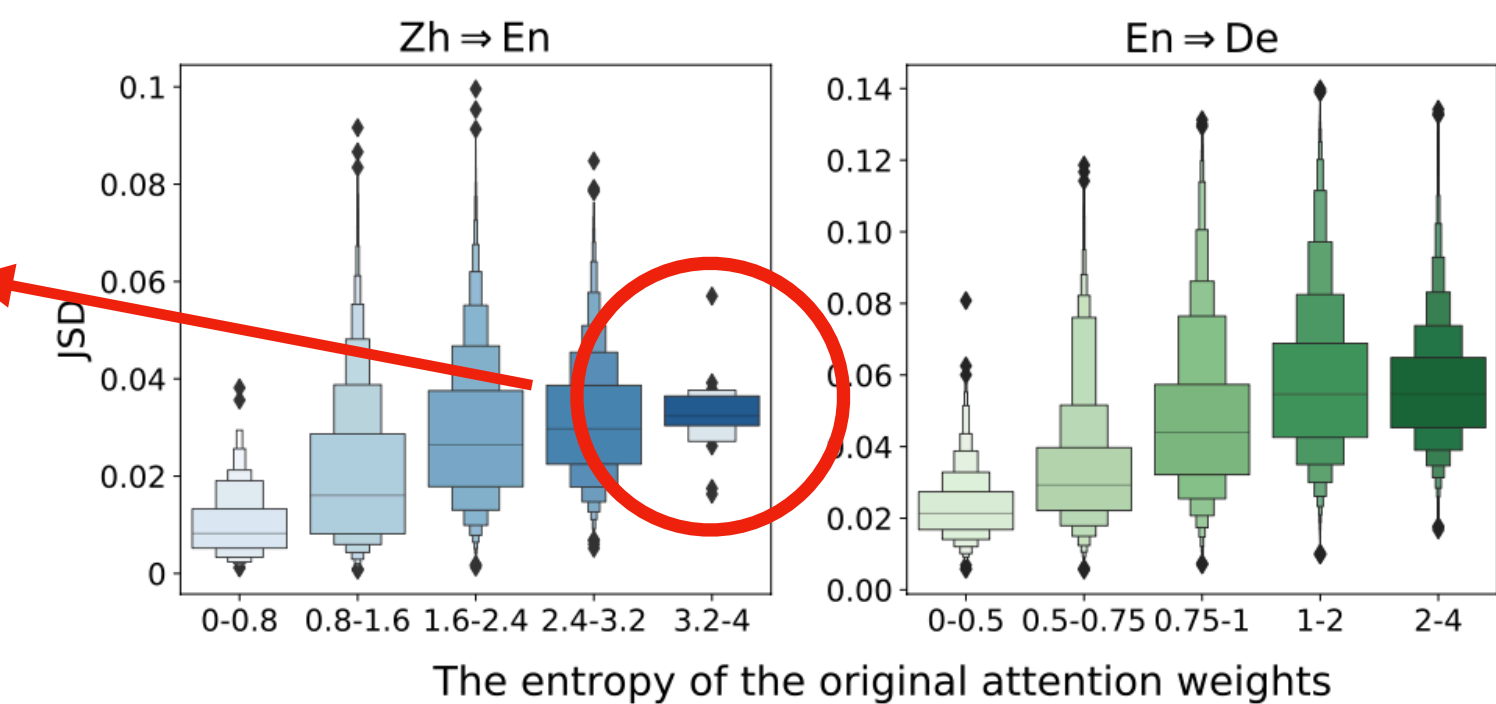
信息熵越大，JSD就越大，矫正程度很高，input需要的rely范围就越分散，这样就容易产生错误

the value is 0.0324 for the attention weights where the entropy is larger than 3.2.

High or low entropy ?



(a) 1-st layer



(b) 6-th layer

Figure 6: The JSD between attention weights before and after calibration at different layers on Zh \Rightarrow En and En \Rightarrow De translation. Note that the overall JSD for each language pair is decided by the hyperparameter α , but the calibration extents of layers are learned by ACN.

Figure 7: The JSD between attention weights before and after calibration with respect to the entropy of original attention distributions.

Analysis

Calibrated attention weights are more dispersed or focused ?

| layer | Zh⇒En | En⇒De |
|-------|----------|----------|
| 1 | + 0.0203 | + 0.1846 |
| 2 | - 0.011 | + 0.0762 |
| 3 | - 0.0023 | + 0.0207 |
| 4 | - 0.0224 | - 0.0336 |
| 5 | - 0.0303 | - 0.0595 |
| 6 | - 0.0083 | - 0.01 |
| All | - 0.0336 | - 0.0224 |

$$\Delta Ent(a_1, a_2) = ent(a_1) - ent(a_2)$$

The low layers generally grasp information from various inputs, while the high layers look for some particular words tied to the model predictions.

Table 5: Entropy differences (ΔEnt) between the original and calibrated attention weights. “+” means the calibrated attention weights are more disperse. “-” indicates attention weights are sharper after calibration.

END