

# Cascaded Head-colliding Attention

Lin Zheng<sup>♠◇</sup> Zhiyong Wu<sup>♠</sup> Lingpeng Kong<sup>♠♣</sup>

<sup>♠</sup>Department of Computer Science, The University of Hong Kong

<sup>◇</sup>School of Data and Computer Science, Sun Yat-sen University

<sup>♣</sup>Shanghai Artificial Intelligence Laboratory

zhenglin6@mail2.sysu.edu.cn, {zywu, lpk}@cs.hku.hk

## Abstract

Transformers have advanced the field of natural language processing (NLP) in many ways. At the heart of the Transformer architecture is the multi-head attention (MHA) mechanism which models pairwise interactions between the elements of the sequence. Despite its massive success, the current framework ignores interactions among different heads, leading to the problem that many of the heads are redundant in practice, which underutilizes the capacity of the model. To improve parameter efficiency, we re-formulate the MHA as a latent variable model from a probabilistic perspective. We present **cascaded head-colliding attention (CODA)** which explicitly models the interactions between attention heads through a hierarchical variational distribution. We conduct extensive experiments and demonstrate that CODA outperforms the transformer baseline, by 0.6 perplexity on `Wikitext-103` in language modeling, and by 0.6 BLEU on `WMT14 EN-DE` in machine translation, due to its improvements on the parameter efficiency.<sup>1</sup>

## 1 Introduction

Transformers (Vaswani et al., 2017) have advanced the field of natural language processing (NLP) on a variety of important tasks, including language modeling (Dai et al., 2019; Baevski and Auli, 2019), language understanding (Devlin et al., 2019; Yang et al., 2019b), and machine translation (Vaswani et al., 2017; Dehghani et al., 2019; Liu et al., 2020). It has also found its place in computer vision (Dosovitskiy et al., 2020), and in intelligent agents (Vinyals et al., 2019) where sequence modeling plays a key role as well. The cornerstone of the transformer architecture is the multi-head attention (MHA) mechanism which models pairwise interactions between the elements of the sequence.

<sup>1</sup>Our implementation is publicly available at <https://github.com/LZhengisme/CODA>.

An attention function can be described as mapping a query and a set of key-value pairs to an output, where the query, keys, values, and output are all vectors. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key. A multi-head attention (MHA) mechanism extends the idea through performing multiple separately parameterized attention functions acting in parallel to contextualize the input representations. Their outputs are then gathered by an affine transformation, allowing the model to jointly attend to information from different representation subspaces at different positions.

Despite its massive success, the current framework ignores the interactions among different heads, leading to the problem that many of the heads are redundant in practice (i.e., attending to the same regions of the sequence), which underutilizes the capacity of the model (Voita et al., 2019; Michel et al., 2019a). At the same time, recent research (Tang et al., 2018; Clark et al., 2019; Voita et al., 2019; Wu et al., 2020, *inter alia*) demonstrates that heads in MHA have the potential to capture distinct information from input sequences, ranging from syntactic and semantic features to alignment information between source and target sentence pairs. These observations suggest that multiple heads should be encouraged to extract complementary information. Therefore, it is highly appealing to take into account the interactions among different attention heads from the perspective of parameter efficiency and the expressiveness of the model.

In this work, we introduce *head-colliding attention* (§3). We formulate MHA as a probabilistic model, where each attention head is represented by a latent variable and all of them *collide* into the observed sequence data (Figure 1a). In this

probabilistic graphical model structure, attention heads work as individual factors to explain the data. Although each factor is independent of each other *a priori*, they interact with each other automatically, conditioning on observations, thanks to the explaining-away effects (Pearl, 1989; Wellman and Henrion, 1993).

The head-colliding attention mechanism introduces new computational challenges in training the model. We will discuss how we tackle these using variational methods (Blei et al., 2017). We propose cascaded head-colliding attention (CODA, Figure 1b). As our main model, CODA adopts a hierarchical variational distribution (Ranganath et al., 2016) to allow both rich head interactions and effective computations (§4).

We validate our method in language modeling and machine translation experiments (§5). CODA outperforms the vanilla MHA transformer on both tasks, on Wikitext-103 by 0.6 perplexity and on WMT14 EN-DE by 0.6 BLEU. Further analysis shows that CODA learns to encourage diversity in different heads (Figure 2) and to promote parameter efficiency when increasing the number of heads (§5.3).

## 2 Background

Multi-head attention (MHA) mechanism plays an important role in modern transformer architecture (Vaswani et al., 2017). It extends the classical attention mechanism by running multiple attention function heads in parallel.

An MHA module is composed of  $h$  identical blocks (usually referred to as attention heads). Each head will generate a hidden state  $\mathbf{H}_i$  based on the input Query, Key and Value matrices, denoted as  $\mathbf{Q}$ ,  $\mathbf{K}$ , and  $\mathbf{V}$  respectively. The hidden states from different heads are then aggregated as the output of the MHA module:  $\sum_{i=1}^n \mathbf{H}_i \mathbf{W}_i^o$ , where  $\mathbf{W}_i^o$  are model parameters.

In the  $i$ -th head, the input matrices  $\mathbf{Q}$ ,  $\mathbf{K}$  and  $\mathbf{V}$  are first linearly projected into different subspace representations  $\tilde{\mathbf{Q}}_i$ ,  $\tilde{\mathbf{K}}_i$ , and  $\tilde{\mathbf{V}}_i$ , based on different learnable parameters. After that, we compute the inner product over all projected queries and keys as the attention logits  $\mathbf{z}_i$ , which are then passed through a row-wise softmax<sup>2</sup> to obtain head attention weights  $\mathbf{a}_i$ :

$$\mathbf{a}_i = \text{softmax}(\mathbf{z}_i) = \text{softmax}(\tilde{\mathbf{Q}}_i \tilde{\mathbf{K}}_i^T). \quad (1)$$

<sup>2</sup>We omit the scaling factor for simplicity.

The final output of a single attention block is the weighted sum of  $\tilde{\mathbf{V}}_i$ :

$$\mathbf{H}_i = \mathbf{a}_i \tilde{\mathbf{V}}_i.$$

As we can see, the core of MHA is to calculate  $\mathbf{a}_i$  in each head. We thus refer to  $\mathbf{a}_i$  as the  $i$ -th attention head.

In sequence prediction tasks, the model takes as input a source sequence of length  $m$  and outputs a target sequence of length  $n$  in an autoregressive manner. It predicts each token  $\mathbf{Y}$  within the target sequence through a categorical distribution  $p_{\text{vanilla}}(\mathbf{Y}|\mathbf{X})$ , where  $\mathbf{X}$  includes the source sequence as well as a previously generated prefix. With respect to an MHA block  $\mathbf{a}_1, \dots, \mathbf{a}_h$ , the model predicts target tokens  $\mathbf{Y}$  by first feeding these heads into a complex non-linear transformation<sup>3</sup> denoted by  $\phi(\cdot)$ , and then passing it through a softmax function over the entire vocabulary. Therefore, the output probability can be written as  $p_{\text{vanilla}}(\mathbf{Y}|\mathbf{X}) = f(\mathbf{a}_1, \dots, \mathbf{a}_h)$ , where

$$f(\mathbf{a}_1, \dots, \mathbf{a}_h) := \text{softmax}(\phi(\mathbf{a}_1, \dots, \mathbf{a}_h)).$$

## 3 Head-colliding Attention

In this section, we introduce *head-colliding attention*. Specifically, we formulate MHA as a probabilistic model, where each attention head is represented by a latent variable. The name reflects a “collider” in the context of probabilistic graphical models (Figure 1a). We will first explain how head-colliding attention permits the modeling of interactions among different heads and then discuss how vanilla MHA can be viewed as a marginalized version of head-colliding attention, which ignores any head interactions.

Considering a single MHA block, we cast each attention head  $\mathbf{a}_i$  as a latent variable. The probability of target  $\mathbf{Y}$  conditioning on input  $\mathbf{X}$  can be obtained by marginalizing over all heads  $\mathbf{A}$  (we denote  $\mathbf{A} := \{\mathbf{a}_1, \dots, \mathbf{a}_h\}$ ):

$$\begin{aligned} p(\mathbf{Y}|\mathbf{X}) &= \int_{\mathbf{A}} p(\mathbf{Y}|\mathbf{A}, \mathbf{X}) p(\mathbf{A}|\mathbf{X}) d\mathbf{A} \\ &= \mathbb{E}_{p(\mathbf{A}|\mathbf{X})} [f(\mathbf{A})]. \end{aligned}$$

$p(\mathbf{A}|\mathbf{X})$  is the joint prior distribution. The corresponding directed graphical model is demonstrated

<sup>3</sup>Since a transformer typically stacks several attentive layers, for an MHA block in some layer, subsequent layers will induce a non-linear transformation  $\phi(\cdot)$  for its attention heads. For instance,  $\phi(\cdot)$  may include several other MHA blocks and feed-forward networks.

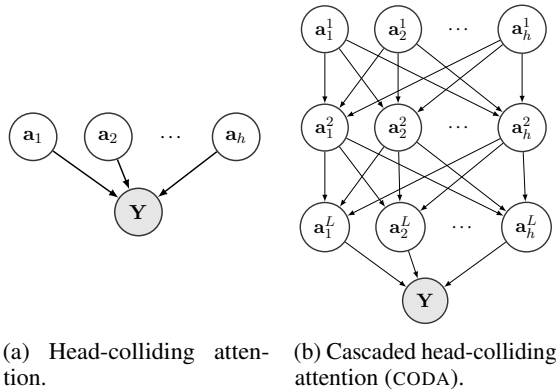


Figure 1: (a) Left: Probabilistic graphical model (PGM) diagram of head-colliding attention. Although each head variable is independent *a priori*, they interact with each other *after* observing targets  $\mathbf{Y}$ , which is referred as explaining-away effect. (b) Right: PGM diagram of a 3-layer cascaded head-colliding attention (CODA).  $\mathbf{a}_i^l$  denotes the  $i$ -th attention head at transformer layer  $l$ . Note that all dependencies from  $\mathbf{X}$  are omitted in these diagrams for simplicity.

in Figure 1a, where the links from different heads collide on the observation variable  $\mathbf{Y}$ . A crucial property of this graphical model is the “explaining-away” effect (Pearl, 1989; Wellman and Henrion, 1993) of attention heads  $\mathbf{A}$  when observing the output  $\mathbf{Y}$ . In other words, if a head  $\mathbf{a}_i$  attends to part of the input which accords well with observation, it immediately discourages other heads from attending to the same part of the input but encourages them to look into complementary information.<sup>4</sup> This mechanism effectively reduces head redundancy and in turn improves parameter efficiency.

**Vanilla vs. head-colliding attention** We now take a closer look at the vanilla MHA (§2). Recall that in vanilla MHA, all attention heads are deterministic. From the perspective of latent variable models, this is computationally equivalent to taking expectations of latent head variables. The output probability distribution  $p_{\text{vanilla}}(\mathbf{Y}|\mathbf{X})$  can then be expressed as:

$$f(\mathbb{E}_{p(\mathbf{a}_1|\mathbf{X})}[\mathbf{a}_1], \dots, \mathbb{E}_{p(\mathbf{a}_h|\mathbf{X})}[\mathbf{a}_h]). \quad (2)$$

This means we are only interested in the individual expectations when using the attention heads in vanilla MHA for predictions. On the contrary,

<sup>4</sup>In other words, if we confirm that some head accords well with the observation, then the probability of other heads should be reduced since there is less need to invoke them, according to Occam’s razor.

in head-colliding attention the distribution of  $\mathbf{Y}$  is defined as:

$$p(\mathbf{Y}|\mathbf{X}) = \mathbb{E}_{p(\mathbf{a}_1, \dots, \mathbf{a}_h|\mathbf{X})} [f(\mathbf{a}_1, \dots, \mathbf{a}_h)].$$

Note the inherent difference of when to take the expectation in vanilla and head-colliding attention. Since  $f(\cdot)$  is a complex non-linear function (§2), these two formulations are not equivalent in general and may have a large gap between the two distributions. Concretely, vanilla MHA ignores any possible interactions among different heads. As indicated in equation 2, it first marginalizes out every single head *before* observing targets – one head will not learn what other heads are attending to despite the fact  $\mathbf{Y}$  is observed. This is why vanilla MHA is prone to redundancy as many previous studies (Voita et al., 2019; Michel et al., 2019a, *inter alia*) discovered. Head-colliding attention, on the other hand, permits rich head interactions due to the expressive non-linear function  $f(\cdot)$  inside the expectation over different latent variables  $\mathbf{a}_1, \dots, \mathbf{a}_h$ . However, the complexity of head interactions also leads to intractability in training the model, which we will discuss in the next section.

## 4 Training Head-colliding Attention

We train the model by performing maximum likelihood estimation. Here, the log marginal likelihood can be expressed as:

$$\log p(\mathbf{Y}|\mathbf{X}) = \log \mathbb{E}_{p(\mathbf{A}|\mathbf{X})} [p(\mathbf{Y}|\mathbf{A}, \mathbf{X})].$$

Unfortunately, this is intractable in general because it requires marginalizing over all possible configurations of attention heads. The standard technique is to use variational inference, which optimizes the log marginal by maximizing its evidence lower bound (called ELBO) (Blei et al., 2017):

$$\begin{aligned} \mathcal{L} &:= \mathbb{E}_{q(\mathbf{A}|\mathbf{X})} \left[ \log \frac{p(\mathbf{Y}|\mathbf{A}, \mathbf{X})p(\mathbf{A}|\mathbf{X})}{q(\mathbf{A}|\mathbf{X})} \right] \quad (3) \\ &= \log p(\mathbf{Y}|\mathbf{X}) - \text{KL}(q(\mathbf{A}|\mathbf{X})||p(\mathbf{A}|\mathbf{X}, \mathbf{Y})) \\ &\leq \log p(\mathbf{Y}|\mathbf{X}), \end{aligned}$$

where  $q(\mathbf{A}|\mathbf{X})$  is the variational distribution<sup>5</sup> over latent variables  $\mathbf{A}$ .  $p(\mathbf{A}|\mathbf{X}, \mathbf{Y})$  is the intractable posterior distribution of all heads given observations  $\mathbf{Y}$  and the input  $\mathbf{X}$ , which encodes the rich

<sup>5</sup>Although the variational distribution  $q$  should depend on target  $\mathbf{Y}$  in principle, such conditioning renders testing difficult since the target information is not available during testing. For this reason, we only consider the source  $\mathbf{X}$  hereafter.

head interactions we desire, as discussed in §3. Therefore, an ideal variational distribution  $q(\mathbf{A}|\mathbf{X})$  should be close to the true posterior  $p(\mathbf{A}|\mathbf{X}, \mathbf{Y})$ . In this case, the samples would accurately reflect the head interactions and the variational distribution would yield a tighter bound to  $\mathcal{L}$  to facilitate the training.

A straight-forward choice of  $q(\mathbf{A}|\mathbf{X})$  is to use the mean-field approximation (Kingma and Welling, 2013):

$$q(\mathbf{A}|\mathbf{X}) = q(\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_h|\mathbf{X}) = \prod_{i=1}^h q(\mathbf{a}_i|\mathbf{X}).$$

However, it has similar drawbacks as the vanilla MHA.<sup>6</sup> The mean-field approximation assumes the independence of different heads and hence the interactions are greatly limited.

Alternatively, one could parameterize  $q(\mathbf{A}|\mathbf{X})$  using an auto-regressive model.<sup>7</sup> Although this is much more expressive, its sequential nature severely slows down training, making it infeasible in practice.

**Cascaded Head-colliding attention** Our solution to this problem is to employ hierarchical structures for head-colliding attention, where interactions among heads could be effectively incorporated into the model (Sønderby et al., 2016; Ranganath et al., 2016).

Conveniently, the hierarchical nature of the transformer architecture offers an effective way of constructing such proposal distributions. Given a transformer with  $L$  layers, we denote the set of all attention heads at layer  $l-1$  and  $l$  as  $\mathbf{A}^{l-1}$  and  $\mathbf{A}^l$ , respectively. Following the bottom-up computation of the transformer, the distribution of  $\mathbf{A}^l$  must rely on the instantiated values of  $\mathbf{A}^{l-1}$ . In this sense,  $\mathbf{A}^{l-1}$  can be seen as the common variables that govern  $\mathbf{A}^l$  (Figure 1b). Formally, we have:

$$q(\mathbf{A}^1, \dots, \mathbf{A}^L|\mathbf{X}) = q(\mathbf{A}^1|\mathbf{X}) \prod_{j=2}^L q(\mathbf{A}^j|\mathbf{X}, \mathbf{A}^{j-1}).$$

Despite the fact that each attention head  $\mathbf{a}_i^l \in \mathbf{A}^l$  at  $l$ -th layer is conditionally independent given  $\mathbf{A}^{l-1}$ , they become dependent when we marginalize  $\mathbf{A}^{l-1}$

<sup>6</sup>Note that the vanilla MHA does not define distributions over heads in its original context. We derive this from the latent-variable perspective.

<sup>7</sup>This works well in our preliminary experience, despite its extremely expensive computational cost.

out. In particular, the marginal distribution of each  $\mathbf{A}^l$  becomes:

$$q(\mathbf{A}^l|\mathbf{X}) = \int_{\mathbf{A}^{l-1}} q(\mathbf{A}^{l-1}|\mathbf{X}) q(\mathbf{A}^l|\mathbf{X}, \mathbf{A}^{l-1}) d\mathbf{A}^{l-1}.$$

This corresponds to an *infinite mixture* of the mean-field distributions  $q(\mathbf{A}^l|\mathbf{X}, \mathbf{A}^{l-1})$  and is able to capture rich head interactions (Ranganath et al., 2016). Our main model adopts this cascaded proposal distribution in figure 1b, and therefore we name it **cascaded head-colliding attention (CODA)**.

The only problem left now is how to specify the conditional distribution  $q(\mathbf{A}^l|\mathbf{X}, \mathbf{A}^{l-1})$  for all  $l = 1, 2, \dots, L$ . We first impose the basic constraints on head values as in vanilla MHA, that is, all head values must range within a simplex  $\Delta^{n-1}$ :

$$\Delta^{n-1} = \{\mathbf{A}^l | \sum_{k=1}^n \mathbf{a}_{i,:k}^l = \mathbf{1}, \forall i = 1, \dots, h\}.$$

Here  $\mathbf{a}_{i,:k}^l$  is the  $k$ -th column of the  $i$ -th attention head at layer  $l$  and  $\mathbf{1}$  denotes the vector of all 1's. For efficient training and inference, we adopt Gaussian-logistic distributions (Blei and Lafferty, 2006; Cohen et al., 2008), which not only satisfy the constraints above but also benefit from the effective reparameterization trick (Kingma and Welling, 2013; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014).

In particular, recall that in vanilla MHA,  $\mathbf{a}_i = \text{softmax}(\mathbf{z}_i) = \text{softmax}(\tilde{\mathbf{Q}}_i \tilde{\mathbf{K}}_i^T)$  (equation 1). We also denote the attention logits at  $l$ -th layer as  $\mathbf{Z}^l := \{\mathbf{z}_1^l, \dots, \mathbf{z}_h^l\}$ . For head  $i$  at layer  $l$ , we first sample from a multivariate Gaussian distribution  $q(\mathbf{z}_{i,j}^l | \mathbf{z}_{i,j}^{l-1})$ <sup>8</sup> and pass the samples into a row-wise softmax function to yield head values:

$$\mathbf{z}_{i,j}^l \sim \mathcal{N}(\mu_{i,j}^l, \Sigma), \quad \mathbf{a}_{i,j}^l = \text{softmax}(\mathbf{z}_{i,j}^l),$$

where  $\mathbf{z}_{i,j}^l$  and  $\mathbf{a}_{i,j}^l$  represent the  $j$ -th row of the  $i$ -th attention logit and attention head at layer  $l$  respectively.

To explicitly model hierarchical structures among attention heads, we propose to add a direct connection between attention heads at adjacent layers (Figure 1b). Such connections offer direct access to the information of attention in the previous layer. Specifically, for each head  $i$  at layer  $l$  we

<sup>8</sup>We only explicitly define the attention logit  $\mathbf{z}$  as random variables, while the distribution of heads  $\mathbf{a}$  is induced via a deterministic transformation (*i.e.*, softmax function) of  $\mathbf{z}$ ; therefore it suffices to build dependencies between attentive logits instead.



set the mean  $\mu_i^l$  as the sum of two parts:

$$\mu_i^l = \underbrace{\tilde{Q}_i \tilde{K}_i^T}_{\text{vanilla MHA}} + \underbrace{\sigma_i(\mathbf{Z}^{l-1})}_{\text{direct connection}}, \quad (4)$$

where  $\sigma_i(\cdot)$  is a two-layer multilayer perceptron (MLP) to fuse information from different heads  $\mathbf{Z}^{l-1}$  (see the cascading connections in Figure 1b for an illustration). We set the covariance matrix  $\Sigma$  to the identity matrix for all attentive logits. We give the prior the same form as the variational posterior and parameters are shared between  $q(\mathbf{A}^1, \dots, \mathbf{A}^L | \mathbf{X})$  and  $p(\mathbf{A}^1, \dots, \mathbf{A}^L | \mathbf{X})$  for our objective (equation 3). With the help of parameter sharing, the KL term in equation 3 is also cancelled out due to the identical distributions.<sup>9</sup> This choice works well in practice, where it not only allows CODA to use almost the same amount of parameters as vanilla Transformer, but also eliminates the need to invoke advanced training techniques for amortized variational inference.<sup>10</sup> More details can be found in Appendix A.

## 5 Experiments

We conduct experiments on language modeling and machine translation tasks.

### 5.1 Setup

**Datasets** First, we conducted experiments for token-level language modeling on a large-scale benchmark dataset Wikitext-103 (Merity et al., 2016), which consists of articles from Wikipedia with the token number around 103M/218K/246K for the training/validation/testing splits respectively. The vocabulary size is 267,744.

For machine translation, we consider two standard datasets:

- WMT14 EN-DE (Bojar et al., 2014), which contains about 4.5M/3K/3K sentences pairs for training/validation/testing splits respectively. We follow Ott et al. (2018) and Peng et al. (2020) to preprocess the dataset, and obtain a shared vocabulary between source and target language of around 32K byte pair encoding (BPE, Sennrich et al. (2016)) types.

<sup>9</sup>Therefore, it can also be derived by directly applying the Jensen’s inequality on the log marginal likelihood.

<sup>10</sup>For instance, training a standard variational auto-encoder (VAE) for NLP tasks often suffers from the posterior collapse problem due to the heavy KL regularization (Bowman et al., 2016), where some tricks have to be used to achieve good performance, such as KL annealing, etc.

- IWSLT14 DE-EN (Cettolo et al., 2014). Following standard practice (Edunov et al., 2018; Peng et al., 2020), we pre-process the 160K/7K/7K sentence pairs and build training/validation/testing sets accordingly. This generates a vocabulary of around 9K(7K) BPE types for source(target).

**Implementation details** We implement our model with PyTorch (Paszke et al., 2019) and FairSeq toolkit (Ott et al., 2019). In particular, our model is based on the vanilla transformer architecture (Vaswani et al., 2017). For CODA, we replace all vanilla MHA blocks with the cascaded head-colliding attention, for both self attention and cross attention (if any). In language modeling, we use adaptive input embeddings (Baeviski and Auli, 2019) and set context size to 512 and 480 for training and testing respectively, due to constraints of computational resources. In machine translation, we set beam size to 5 and adopt the hyperparameters from (Peng et al., 2020) for IWSLT14 DE-EN. For WMT14 EN-DE we set beam size to 4, length penalty to 0.6, and average last 10 checkpoints for testing, following Vaswani et al. (2017). Further implementation details can be found in Appendix A.

### 5.2 Main results

The results of language modeling on Wikitext-103 dataset are reported in Table 1. As we can see from the table, CODA barely introduces any additional parameters. However, by taking into account head interactions, CODA significantly outperforms TRANSFORMER by over 0.6 perplexity. For reference, we also report the best setting (denoted by TRANSFORMER<sup>†</sup>) in Baeviski and Auli (2019), which uses a much larger context size (3072/2560 vs. 512/480 for training/testing), CODA still outperforms by a substantial margin of 0.3 perplexity. This indicates that encouraging head interactions can improve parameter efficiency.

To show whether CODA has promoted head interactions and reduced head redundancy, we qualitatively visualize the attention heads in both CODA and TRANSFORMER via heatmaps. Concretely, we compute the Jensen-Shannon Divergence (JSD) between each pair of attention heads at the same layer.

In particular, we assume head values define a categorical distribution in both TRANSFORMER and CODA model to facilitate comparison. That is, an

Model	# Params.	Val. PPL	Test PPL
TRANSFORMER	246.93M	18.35	19.08
TRANSFORMER <sup>†</sup>	246.93M	17.97	18.70
CODA	246.96M	<b>17.81</b>	<b>18.48</b>

Table 1: Validation (Val.) and testing Perplexity (PPL) on Wikitext-103 dataset (lower is better). TRANSFORMER is the base model in Baevski and Auli (2019) with the same context size as CODA (512/480 for training/testing), while TRANSFORMER<sup>†</sup> is the same model but with the best setting in their paper, which uses much larger context size (3072/2560 respectively); the result for TRANSFORMER<sup>†</sup> is as reported in Baevski and Auli (2019).

attentive head  $\mathbf{a}_i$  induces  $n$  categorical distributions for each query position. For the  $j$ -th distribution, it indicates how the  $j$ -th target position attends to all  $m$  source positions and is denoted by  $p(x|\mathbf{a}_{i,j})$ . For two heads  $i$  and  $i'$ , we first compute their average distribution as

$$m := \frac{p(x|\mathbf{a}_{i,j}) + p(x|\mathbf{a}_{i',j})}{2}$$

Then the JSD value between the  $i$ -th and  $i'$ -th attention head is computed by summing all of  $n$  induced distributions:

$$\sum_{j=1}^n \frac{1}{2} (\text{KL}(p(x|\mathbf{a}_{i,j})||m) + \text{KL}(p(x|\mathbf{a}_{i',j})||m))$$

We average computed JSDs for all validation samples. Note that a larger JSD value (darker color) indicates that two heads are behaving more differently (i.e. less redundancy between them), and vice versa.

As shown in Figure 2, JSD heatmaps in CODA are clearly darker than those in TRANSFORMER. This suggests that CODA permits richer head interactions, which fosters different heads to communicate with each other and encourages them to become complementary. Consequently, our model effectively reduces head redundancy in MHA and improves parameter-efficiency.

The results on IWSLT14 DE-EN and WMT14 EN-DE datasets are shown in Table 2. We see that CODA exhibits clear improvements over TRANSFORMER: a 1.1 point gain in BLEU on IWSLT14 DE-EN dataset and a 0.6 BLEU improvement on WMT14 EN-DE dataset. Despite such significant gains over the baseline, CODA only introduce very few additional parameters (e.g., 0.03% extra parameters on IWSLT14 DE-EN). This, again, shows

Model	IWSLT14 DE-EN		WMT14 EN-DE	
	# Params.	BLEU	# Params.	BLEU
TRANSFORMER	39.47M	34.5	60.92M	27.4
CODA	39.48M	<b>35.6</b>	60.94M	<b>28.0</b>

Table 2: Performance of TRANSFORMER and CODA on IWSLT14 DE-EN and WMT14 EN-DE datasets.

that CODA is more parameter efficient than vanilla Transformer due to the cascaded head-colliding attention we proposed. Similar to experiments on language modeling, we also visualize the head behaviors to measure attentive head interactions (See Figure 5 and Figure 6 in Appendix B), where we observe similar phenomena on translation tasks. Specifically, different heads in CODA are often complementary to each other and focus on quite different regions of sequences, rather than becoming redundant or even identical as observed in TRANSFORMER models.

### 5.3 Analysis: the effect of the number of attention heads

Despite one would hope increasing the head number in MHA leads to a free-ride in achieving better performance, in practice it is often not the case as vanilla MHA suffers from the problem of parameter redundancy. Following Vaswani et al. (2017), we vary the number of attention heads (4,8,16,32), but keep the amount of computation constant. Our results on IWSLT14 DE-EN are shown in Table 3. We observe that the translation quality of baseline transformer (which uses vanilla MHA as its main building blocks) decreases almost linearly when increasing number of attention heads (Figure 3), which agrees with previous studies (Vaswani et al., 2017; Voita et al., 2019; Michel et al., 2019b).

Intuitively, since the total number of parameters in the model remains unchanged, more heads indicate that the number of parameters allocated to each head is reduced, which limits the representational power of every single attention head. Due to the independence assumption between the heads, many of them tend to focus on similar regions of the sequence, leading to a great waste of modeling capacity.

In the case of CODA, we observe better BLEU scores in response to the increasing head number. Rich interactions in CODA could encourage different heads to cover broader regions of input sequence, which in turn offers more useful information for training. The perplexity (PPL) reflects

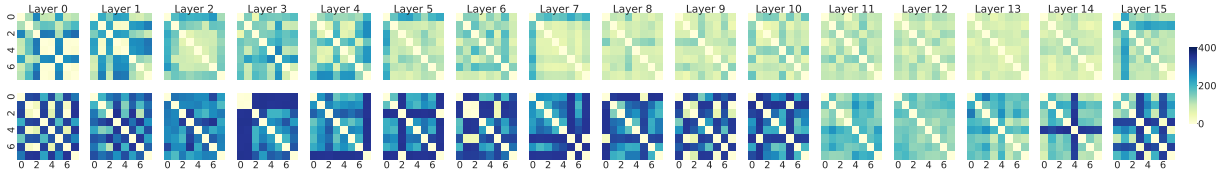


Figure 2: Jensen-Shannon Divergences (JSD) for each pair of attention heads at all 16 layers on Wikitext-103 validation dataset. Top: JSD heatmap of attention heads from TRANSFORMER model; Bottom: JSD heatmap of attention heads from CODA. Columns represent different layers of both models. The darker color implies a larger divergence between two heads and in turn less redundancy.

# heads	BLEU		PPL	
	TRANSFORMER	CODA	TRANSFORMER	CODA
4	34.53	<b>35.65</b>	4.95	<b>4.64</b>
8	34.35	<b>35.74</b>	5.04	<b>4.54</b>
16	33.91	<b>35.84</b>	5.15	<b>4.55</b>
32	33.17	<b>35.96</b>	5.37	<b>4.52</b>

Table 3: Left: BLEU scores on test dataset for TRANSFORMER and CODA at different numbers of attention heads; Right: Perplexity on validation dataset for TRANSFORMER and CODA at different numbers of attention heads.

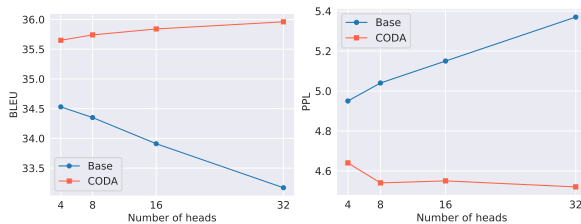


Figure 3: Left: BLEU scores on test dataset for base transformers and CODA under different number of attention heads (higher is better); Right: Perplexity on validation dataset for base transformers and CODA under different number of attention heads (lower is better).

similar trends. The coordination between different heads in CODA greatly improves the model’s parameter efficiency.

## 5.4 Ablation analysis

In this section, we present an ablation study to investigate effects of different components in CODA. Concretely, we compare four models on the IWSLT14 DE-EN machine translation task: (i) the full model CODA, (ii) a variant of CODA ablating the cascaded structure (§4), (iii) a variant of CODA without using head-colliding attention (§3) and (iv) the baseline TRANSFORMER model.

In more details, for model (ii), we remove the second term in equation 4, which turns off the direct cascading structure, despite still being a proper

hierarchical latent variable model<sup>11</sup>. In model (iii), attention heads are deterministic (instead of being latent variables) as in vanilla Transformers, but cascading connections are incorporated. We observe its close connection with the recently proposed REALFORMER (He et al., 2020), a TRANSFORMER model that adds a residual connection between attention logits at adjacent layers. Since in model (iii) all attention heads are deterministic, it is unnecessary to fuse different heads (see §4). In this case, we simply implement model (iii) as a REALFORMER (and thus referred to as REALFORMER hereafter) to demonstrate the effect of cascading-like structures more clearly.<sup>12</sup>

We report BLEU score for translation quality, and the Jensen-Shannon Divergences (JSD) averaged over all heads pairs of all MHA blocks for quantitative evaluation of head interactions. As demonstrated in Table 4 and Figure 4, even without cascading connections for explicit hierarchical structures, head-colliding attention has the ability (albeit limited) to induce reasonable correlations among different heads, reflected in the average JSD. This is due to the explaining-away effects and the native hierarchical structure in the transformers, as discussed in §3. In CODA, because individual heads have access to the other heads from a probabilistic perspective, they are more prone to offering complementary information for each other to jointly explain the observed data. This effect is further enhanced when cascading connections are added to the model. In contrast, if we simply incorporate such cascading connections into a vanilla TRANSFORMER model, we found it does not significantly

<sup>11</sup>Note that the first term  $\tilde{Q}_i \tilde{K}_i^T$  in equation 4 also depends on the instantiated value of  $z_{i,j}^{l-1}$ , which induces an implicit hierarchical dependency for attention between adjacent layers.

<sup>12</sup>The main difference between residual connections in REALFORMER and cascading connections in CODA is that, the former directly performs a *head-wise* addition of previous-layer attention logits; in contrast, our cascading connection makes use of an MLP  $\sigma(\cdot)$  to mix different attention heads, which enhances head interactions for CODA.

Model	Avg. JSD	BLEU
CODA	13.72	35.65
CODA-CS	11.24	35.17
REALFORMER (He et al., 2020)	8.53	35.01
TRANSFORMER	7.11	34.53

Table 4: The average JSD and BLEU scores with different model configurations. CODA-CS indicates the ablation of the cascading structures from the full model (i.e., simply replacing all MHA blocks of base transformer with head-colliding attention); REALFORMER is a recently proposed TRANSFORMER model that has cascading-like structures but still views each head as a deterministic value rather than latent variables.

encourage head interactions and only improves the baseline marginally. In this case, the performance improvement might be mainly due to residual connections, which are often considered to be effective in facilitating training (He et al., 2016). Interestingly, we note a positive correlation between average JSD and BLEU, suggesting that encouraging complementary attention heads may help improve translation quality.

## 6 Related Work

Attention mechanisms were first applied to recurrent networks in (Bahdanau et al., 2014). It was then extended to multi-head attention (MHA) and became the key component in transformer architectures (Vaswani et al., 2017).

To study the utility of multiple attention heads, Voita et al. (2019) focused on identifying individual contributions of each attention head. Michel et al. (2019a) conducted extensive experiments to demonstrate that pruning out most heads after training does not lead to a drop in performance during inference. You et al. (2020) further revealed that replacing learnable attention heads with samples from fixed Gaussian distributions can achieve almost the same performance as original models. Additionally, Behnke and Heafield (2020) proposed to iteratively prune attention heads during training based on the lottery ticket hypothesis. These works indicate that there is a lot of head redundancy in the MHA transformer architectures.

Instead of pruning unnecessary parameters and down-sizing transformer models, there are also works that propose to improve parameter efficiency in transformers. For instance, Li et al. (2018) introduced a regularization term to explicitly promote diversity among different heads. Yang et al. (2019a) proposed to use convolutional kernels to

capture correlations among not only local windows of sequences, but also different heads. An et al. (2020) considered each head as a sample from the same distribution, and presented a sampling algorithm that avoids samples from collapsing into local modes. It hence explicitly encouraged the repulsiveness in MHA. Besides, MAE (Peng et al., 2020) converted a vanilla MHA to a mixture-of-experts model, where each expert component activates only a subset of attention heads. With learned probabilities, different experts could be specialized on different inputs. Different from these works, CODA does not explicitly promote head diversity nor specialize different heads. Instead, we focus on studying head interactions from a probabilistic perspective, which reveals the close connection between vanilla MHA and CODA.

Another research line relating to our work is to incorporate latent variables into attention modules. Xu et al. (2015) investigated the connection between vanilla deterministic single-head attention and its stochastic counterpart. Deng et al. (2018) explored this further and proposed to use variational inference techniques for training the model. They considered both cases of discrete and continuous latent variables. Bayesian attention modules (Fan et al., 2020) introduced continuous latent distributions for attention that are amenable to reparameterization tricks. Our work is different from them in that we mainly investigate the MHA mechanism and aim to improve parameter-efficiency by recovering potential interactions among different heads, which are ignored in vanilla MHA.

Concurrently, He et al. (2020) proposed to add residual connections between attention scores at adjacent layers, similar to our cascading connections. Nevertheless, our motivation for using the cascaded structure is quite different: we aim to construct direct hierarchical dependencies for latent variable models, while He et al. (2020) is mainly motivated to improve transformer architectures and obtain performance gains.

## 7 Conclusion and Future Work

We present CODA by re-formulating the multi-head attention (MHA) as a latent variable model from a probabilistic perspective. CODA explicit models of the interactions among attention heads through a hierarchical variational distribution. We conduct extensive experiments and demonstrate that CODA outperforms the transformer baseline in language



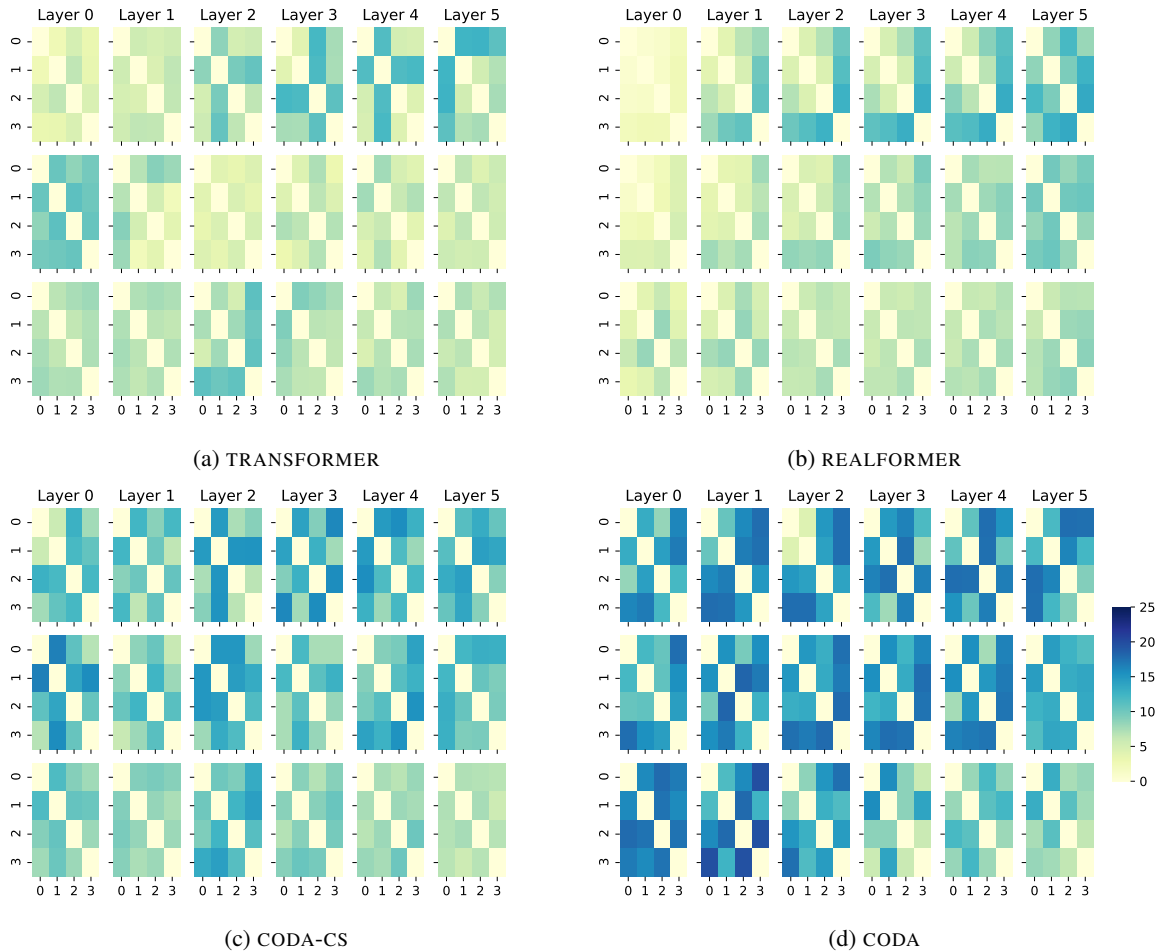


Figure 4: Jensen-Shannon Divergences (JSD) for each pair of attention heads at the same layer on IWSLT14 DE-EN dataset for TRANSFORMER, REALFORMER, CODA-CS and CODA model respectively. Each row indicates different kinds of attention, including encoder self-attention, decoder self-attention and decoder-encoder cross attention (from top to bottom), respectively; and each column indicates average JSD scores at different layers.

modeling and machine translation. The analysis shows that CODA learns to encourage the diversity in different heads and to promote parameter efficiency when increasing the number of heads. In this framework, we will be able to impose explicit constraints or regularization on different attention heads in a principal way (e.g. informative priors that promote diversity). Besides, we can also consider more expressive (data-driven) variational distributions. We leave these as the future work. Our code is publicly available at <https://github.com/LZhengisme/CODA>.

## Acknowledgments

We thank the anonymous reviewers whose suggestions helped clarify this work. This research was supported in part by the University of Hong Kong Research Committee under account 104006039.111994.14200.301.01.

## References

- Bang An, Jie Lyu, Zhenyi Wang, Chunyuan Li, Changwei Hu, Fei Tan, Ruiyi Zhang, Yifan Hu, and Changyou Chen. 2020. **Repulsive attention: Rethinking multi-head attention as Bayesian inference**. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 236–255, Online. Association for Computational Linguistics.
- Alexei Baevski and Michael Auli. 2019. **Adaptive input representations for neural language modeling**. In *International Conference on Learning Representations*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.
- Maximiliana Behnke and Kenneth Heafield. 2020. **Loosing heads in the lottery: Pruning transformer attention in neural machine translation**. In *Proceedings of*

- the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 2664–2674, Online. Association for Computational Linguistics.
- David Blei and John Lafferty. 2006. Correlated topic models. *Advances in neural information processing systems*, 18:147.
- David M Blei, Alp Kucukelbir, and Jon D McAuliffe. 2017. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.
- Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, et al. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the ninth workshop on statistical machine translation*, pages 12–58.
- Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. [Generating sentences from a continuous space](#). In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21, Berlin, Germany. Association for Computational Linguistics.
- Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. 2014. Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation, Hanoi, Vietnam*, volume 57.
- Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert’s attention. *arXiv preprint arXiv:1906.04341*.
- Shay Cohen, Kevin Gimpel, and Noah A Smith. 2008. Logistic normal priors for unsupervised probabilistic grammar induction. *Advances in Neural Information Processing Systems*, 21:321–328.
- Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. [Transformer-XL: Attentive language models beyond a fixed-length context](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.
- Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. [Universal transformers](#). In *International Conference on Learning Representations*.
- Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. 2018. Latent alignment and variational attention. *Advances in Neural Information Processing Systems*, 31:9712–9724.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. [BERT: Pre-training of deep bidirectional transformers for language understanding](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.
- Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc’Aurelio Ranzato. 2018. [Classical structured prediction losses for sequence to sequence learning](#). In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.
- Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. 2020. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 33.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. 2020. [RealFormer: Transformer Likes Residual Attention](#). *arXiv e-prints*, page arXiv:2012.11747.
- Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.
- Jian Li, Zhaopeng Tu, Baosong Yang, Michael R. Lyu, and Tong Zhang. 2018. [Multi-head attention with disagreement regularization](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2897–2903, Brussels, Belgium. Association for Computational Linguistics.
- Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. 2020. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

- Paul Michel, Omer Levy, and Graham Neubig. 2019a. Are sixteen heads really better than one? In *Advances in Neural Information Processing Systems*, pages 14014–14024.
- Paul Michel, Xian Li, Graham Neubig, and Juan Pino. 2019b. On evaluation of adversarial perturbations for sequence-to-sequence models. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3103–3114, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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.
- Myle Ott, Sergey Edunov, David Grangier, and Michael Auli. 2018. Scaling neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 1–9, Brussels, Belgium. Association for Computational Linguistics.
- 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. 2019. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.
- J. Pearl. 1989. Probabilistic reasoning in intelligent systems - networks of plausible inference. In *Morgan Kaufmann series in representation and reasoning*.
- Hao Peng, Roy Schwartz, Dianqi Li, and Noah A. Smith. 2020. A mixture of h - 1 heads is better than h heads. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6566–6577, Online. Association for Computational Linguistics.
- Rajesh Ranganath, Dustin Tran, and David Blei. 2016. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333.
- Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1278–1286.
- 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.
- Casper Kaae Sønderby, Tapani Raiko, Lars Maaløe, Søren Kaae Sønderby, and Ole Winther. 2016. Ladder variational autoencoders. In *Advances in Neural Information Processing Systems*, volume 29, pages 3738–3746. Curran Associates, Inc.
- Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. 2013. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, pages 1139–1147.
- Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. 2018. Why self-attention? a targeted evaluation of neural machine translation architectures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4263–4272, Brussels, Belgium. Association for Computational Linguistics.
- Michalis Titsias and Miguel Lázaro-Gredilla. 2014. Doubly stochastic variational bayes for non-conjugate inference. In *Proceedings of the 31st International Conference on Machine Learning*, volume 32, pages 1971–1979.
- 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 *Advances in neural information processing systems*, pages 5998–6008.
- Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al. 2019. Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, 575(7782):350–354.
- 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.
- Michael P. Wellman and M. Henrion. 1993. Explaining ‘explaining away’. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:287–292.
- Zhiyong Wu, Yun Chen, Ben Kao, and Qun Liu. 2020. Perturbed masking: Parameter-free probing for analyzing and interpreting BERT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4166–4176, Online. Association for Computational Linguistics.

- Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057.
- Baosong Yang, Longyue Wang, Derek F. Wong, Lidia S. Chao, and Zhaopeng Tu. 2019a. [Convolutional self-attention networks](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4040–4045, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. [Xlnet: Generalized autoregressive pretraining for language understanding](#). In *Advances in Neural Information Processing Systems*, volume 32, pages 5753–5763. Curran Associates, Inc.
- Weiqiu You, Simeng Sun, and Mohit Iyyer. 2020. [Hard-coded Gaussian attention for neural machine translation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7689–7700, Online. Association for Computational Linguistics.



## A Implementation details

For the  $\sigma$  network, it consists of a 2-layer MLP with LeakyRelu non-linear activation and a residual link from the input. It is a rather small network and only accounts for 0.01-0.02% of the total parameters. Recall that the number of attention heads is denoted by  $h$ , the source and target length is  $m$  and  $n$  respectively, and the batch size is denoted by  $b$ . The hidden size is set to  $\alpha * h$ , where we select  $\alpha$  from  $\{2, 4, 8\}$  based on the validation set. Note that the additionally introduced number of parameters is negligible compared to the model size, accounting for only 0.01-0.02% of the total parameters. Since we often represent the attention scores (or logits)  $\mathbf{z}$  as a multi-dimensional tensor with shape  $(b, h, n, m)$ , we first transpose it to shape  $(b, m, n, h)$  and feed it into the  $\sigma$  network. It then outputs  $h$  values so that each component  $\sigma_i$  computes the fused information from all previous layer’s attention heads. By adding its output to the current layer’s attention logits, we could effectively construct a direct cascading connection for our hierarchical proposal. Note that  $\sigma$  network is neither shared among different heads nor different layers.

### A.1 Machine translation

For WMT14 EN-DE, the transformer-base architecture in Vaswani et al. (2017) is used, where both the encoder and decoder consist of 6 layers with hidden size 512. For MHA blocks at each layer, the number of attention heads is set to 8 with the dimension of hidden layer representations being 512; For feed forward networks, the hidden size is set to 2048. The rate of dropout is set to 0.1. For training, we follow the same setup as in Vaswani et al. (2017), including that label smoothing with rate 0.1, the Adam optimizer (Kingma and Ba, 2014) is used for optimization, the inverse square root scheduling is utilized for learning rate and the number of warm-up steps is set to 4000.

For IWSLT-14, we follow the configuration of hyper-parameters in Fairseq package<sup>13</sup>. In details, it mostly follows the same architecture and training setup as above, except that it uses a smaller feed forward network with hidden dimension 1024, a larger dropout rate 0.3 and less attention heads 4.

For both datasets, we apply a compound split post-processing to facilitate comparison. Addition-

<sup>13</sup><https://github.com/pytorch/fairseq/tree/master/examples/translation>

ally, we use activation dropout with rate 0.1 for all used models on both datasets as we find it helps our model converge better.

### A.2 Language modeling

For Wikitext-103, we base our model on Baevski and Auli (2019) with the same hyperparameter configuration and training setup. The model architecture consists of 16 transformer layers, where it uses adaptive input representations, 8 heads for each MHA block, dropout rate of 0.3, hidden dimension of 1024, and hidden size of 4096 for feed forward networks. For training, Nesterov’s accelerated gradient (NAG) method (Sutskever et al., 2013) is used with gradient norm clipping and a cosine learning rate schedule<sup>14</sup>.

## B Additional experimental results

Figure 5 and Figure 6 visualize head interactions within TRANSFORMER and CODA on IWSLT14 DE-EN and WMT14 EN-DE translation tasks respectively.

<sup>14</sup>More details can be found in Baevski and Auli (2019) and the training script based on Fairseq codebase: [https://github.com/pytorch/fairseq/blob/master/examples/language\\_model/README.adaptive\\_inputs.md](https://github.com/pytorch/fairseq/blob/master/examples/language_model/README.adaptive_inputs.md).

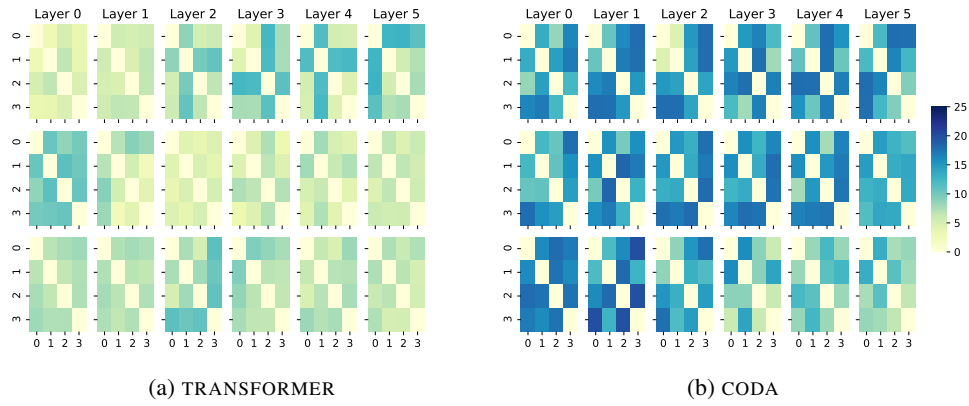


Figure 5: Jensen-Shannon Divergences (JSD) for each pair of attention heads at the same layer on IWSLT14 DE-EN validation dataset, which are evaluated on both TRANSFORMER model and CODA. Each row indicates different kinds of attention, including encoder self-attention, decoder self-attention and decoder-encoder cross attention (from top to bottom), respectively; and each column indicates average JSD scores at different layers.

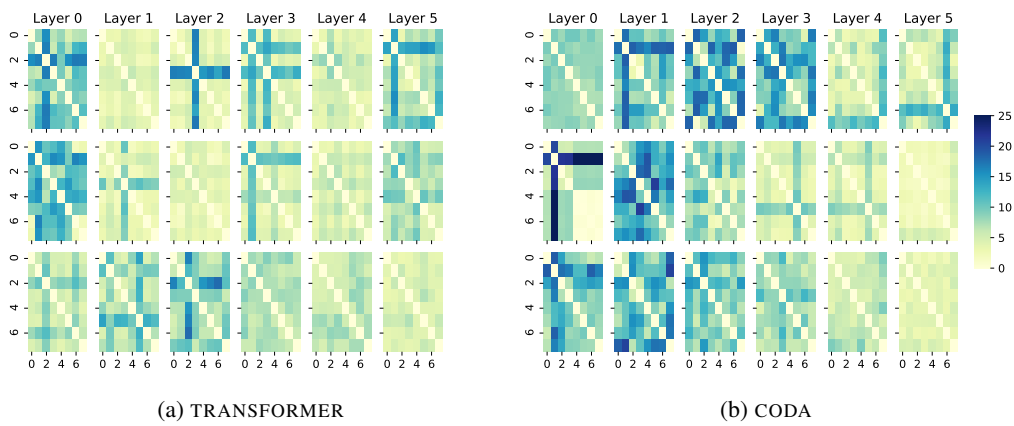


Figure 6: Jensen-Shannon Divergences (JSD) for each pair of attention heads at the same layer on WMT14 EN-DE dataset, which are evaluated on both TRANSFORMER model and CODA. Each row indicates different kinds of attention, including encoder self-attention, decoder self-attention and decoder-encoder cross attention (from top to bottom), respectively; and each column indicates average JSD scores at different layers.