CROSSFORMER: TRANSFORMER WITH ALTERNATED CROSS-LAYER GUIDANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Transformers with stacked attention layers have achieved state-of-the-art results on a wide range of tasks related to discrete sequences. Significant work has been done to better understand or interpret the capabilities of Transformer, which is often massively over-parameterized and prone to overfitting. There exist intensive interactions between Transformer layers, where the information from higher layers can and do distill the information from lower layers. This motivates us to inject a cross-layer inductive bias that not only uses higher layers, which are closer to the training objective, to guide lower ones, but also provides regularization customized to the stacked structure of Transformer. To this end, we propose Crossformer that either regularizes the differences between specific states of two adjacent layers, or directly imposes alternated states sharing between all adjacent layers. Crossformer with states sharing not only provides the desired cross-layer guidance and regularization but also reduces the memory requirement. It is simple to convert a Transformer-based model to a Crossformer-based one. On a variety of neural machine translation tasks, we show that our method outperforms Transformer models while being more memory-efficient. We further demonstrate the general applicability and stability of Crossformer on visual question answering, graph node classification, and significantly deeper models, showing the great potential of incorporating our method into various Transformer-related tasks.

1 INTRODUCTION

There has been significant recent interest in Transformer (Vaswani et al., 2017), which has become the dominant neural network architecture for natural language processing (NLP) and related sequence modeling tasks. Transformer was born with a stacked structure with multiple layers, each of which has a trainable multi-head attention that uses a query-key-value decomposition to capture complex dependencies among sequence tokens, followed by a feed-forward network (FFN) to learn wider representations. Taking advantages of the attention mechanism (Bahdanau et al., 2014) and stacked structure, Transformer and related models have achieved great success in a wide variety of research areas, such as language representations (Devlin et al., 2018; Lan et al., 2019), neural machine translation (Dehghani et al., 2018; Edunov et al., 2018), computer vision (Dosovitskiy et al., 2020; Touvron et al., 2021), graph analysis (Veličković et al., 2017; Yun et al., 2019), and multi-modal learning (Yu et al., 2019; Lee et al., 2020; Cornia et al., 2020).

Significant work has been done to better understand or interpret the capabilities of Transformer layers. Lu et al. (2019) provide a novel perspective that the Transformer layers can be naturally interpreted as a numerical ODE solver for a first-order convection diffusion equation in a multi-particle dynamic system. The work on model probing (Tenney et al., 2019a;b; Liu et al., 2019) shows that syntactic and semantic features can be represented at different Transformer layers. Tenney et al. (2019a) explicitly reveal that the Transformer model often revises the ambiguous representations in its lower layers to produce more definite representations in its higher layers. Liu et al. (2019) further demonstrate that the best performing layer of Transformer is usually near the middle or top.

A general observation of these studies is that while higher-layer representations in Transformer are synthesized from lower-layer ones, they are closer and less ambiguous to the training objective (Tenney et al., 2019a). This view motivates us to consider purposely injecting a cross-layer inductive bias into the Transformer structure that explicitly encourages its lower layers to be guided by higher layers, which may help better expose its lower-layer parameters to the training objective.

Our motivation of injecting a cross-layer guidance also coincides with the call for proper regularization in Transformer, which, despite its effectiveness, is often massively over-parameterized and hence prone to overfitting, especially when there are many stacked layers. Conventional regularization methods, such as applying weight decay on weight matrices (Krogh & Hertz, 1992), data augmentation on embeddings (Sennrich et al., 2015), and dropout on neurons (Srivastava et al., 2014), are commonly adopted to alleviate the overfitting issues (Wu et al., 2021). Different from these conventional regularization methods that are not customized for a stacked structure, injecting a cross-layer inductive bias could be well suited to regularize the stacked structure of multiple closely-related layers. As a proof of concept, we have tried injecting a simple cross-layer inductive bias by pulling the key matrix in multi-head attention towards the query matrix of the layer above under an $L1$ or $L2$ regularization, which allows the higher-layer states to provide a soft-guidance to the lower-layer states. Surprisingly, on multiple Transformer benchmark tasks, we consistently observe tangible improvements introduced by this simple cross-layer guidance on multi-head attention.

Motivated by this observation, we put forward two specific implementations of Crossformer built on top of existing Transformer. The first one, Crossformer-SG (soft-guide), introduces cross-layer regularization as an inductive bias to softly guide the lower-layer states using the high-layer information. Using multi-head attention as an example, Crossformer-SG introduces a regularizer to minimize the discrepancy between the key matrix in each layer and the query matrix one layer above. For FFN, Crossformer-SG alternatively regularizes each of the two bottleneck matrices in adjacent layers. We continue to propose our second implementation, Crossformer-HG (hard-guide), by unifying the two alignment state matrices in the regularizer into one single shared matrix. This not only drives the regularizer to zero, providing a straightforward way of imposing cross-layer structural regularization, but also reduces the parameter number in the model. Figure 1 provides a clear illustration of the Crossformer blocks.

Crossformer-HG modifies multi-head attention by sharing the query of the current layer as the key of the lower layer, and modifies FFN by utilizing the weight from the current layer as the weight in the lower layer within the FFN. The learned information from higher layers can and do distill that from lower layers. With a generic architecture, Crossformer can convert any existing Transformer models while maintaining the inherent advantages of conventional Transformer, such as efficiency and being simple to optimize. One great advantage of this type of regularization is that it is free of additional resource requirements and extra model designs. Through cross-layer state sharing, this structure can not only achieve stronger performance but also occupy less computational memory.

Our method of introducing cross-layer guidance, which requires only a few modifications to Transformer, is simple to implement, stable to train, and maintains good scalability, thereby making it attractive for large-scale deep learning applications. We evaluate the proposed method on a broad range of tasks, including various neural machine translations tasks with very deep models, visual question answering, and graph attention networks. We show that the proposed Crossformer, despite taking less computational memory, consistently outperforms the baseline models. We further demonstrate the stability and generalizability of the proposed method in very deep models.

Our main contributions are summarized as follows:

- Present the first study of cross-layer regularization in Transformers via guiding lower layers with higher layer information.
- Propose a method which requires only a few modifications to standard Transformer model, is stable to train, and maintains good scalability with less computational memory.
- Achieve consistent gains on various neural machine translations tasks and demonstrate the general applicability in visual question answering, graph node classification, and very deep Crossformer models.

2 CROSSFORMER: TRANSFORMER WITH CROSS-LAYER GUIDANCE

In this section, we first inject a cross-layer inductive bias into Transformer by first introducing a soft cross-layer guidance to regularize the differences between specific states of adjacent layers, and

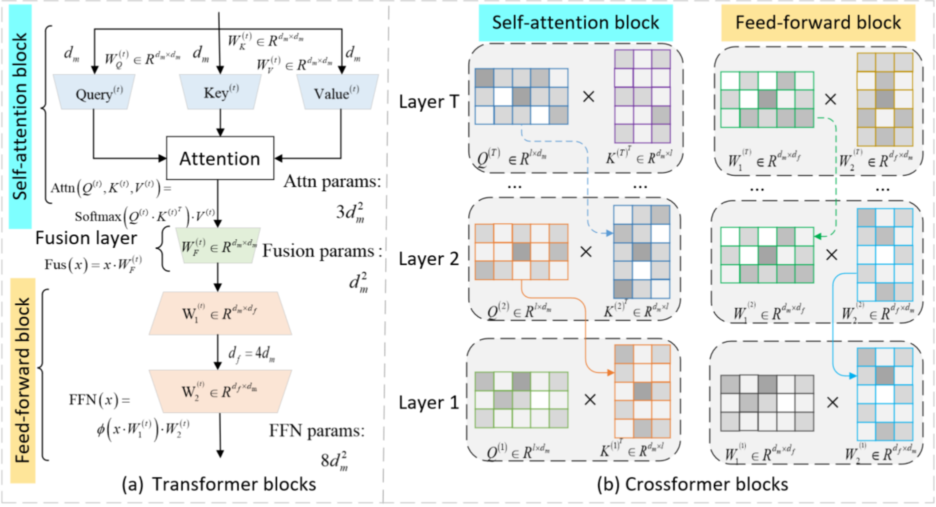

Self-attention block
d_m
$W_Q^{(t)} \in R^{d_m \times d_m}$
d_m
$W_K^{(t)} \in R^{d_m \times d_m}$
$W_V^{(t)} \in R^{d_m \times d_m}$
d_m
Query$^{(t)}$
Key$^{(t)}$
Value$^{(t)}$
Attention
$\mathrm{Attn}(Q^{(t)}, K^{(t)}, V^{(t)}) =$
$\mathrm{Softmax}(Q^{(t)} \cdot K^{(t)^T}) \cdot V^{(t)}$
Attn params:
$3d_m^2$
Fusion layer
$\mathrm{Fus}(x) = x \cdot W_F^{(t)}$
$W_F^{(t)} \in R^{d_m \times d_m}$
Fusion params :
d_m^2
Feed-forward block
$W_1^{(t)} \in R^{d_m \times d_f}$
$d_f = 4d_m$
$W_2^{(t)} \in R^{d_f \times d_m}$
$\mathrm{FFN}(x) =$
$\phi(x \cdot W_1^{(t)}) \cdot W_2^{(t)}$
FFN params:
$8d_m^2$
(a) Transformer blocks
Self-attention block
Feed-forward block
Layer T
$Q^{(T)} \in R^{l \times d_m}$
$K^{(T)^T} \in R^{d_m \times l}$
$W_1^{(T)} \in R^{d_m \times d_f}$
$W_2^{(T)} \in R^{d_f \times d_m}$
Layer 2
$Q^{(2)} \in R^{l \times d_m}$
$K^{(2)^T} \in R^{d_m \times l}$
$W_1^{(2)} \in R^{d_m \times d_f}$
$W_2^{(2)} \in R^{d_f \times d_m}$
Layer 1
$Q^{(1)} \in R^{l \times d_m}$
$K^{(1)^T} \in R^{d_m \times l}$
$W_1^{(1)} \in R^{d_m \times d_f}$
$W_2^{(1)} \in R^{d_f \times d_m}$
×
...
(b) Crossformer blocks

Figure 1: (a) is the standard Transformer block at t-th layer. (b) is the proposed T layers Crossformer's self-attention (left) block and feed-forward block (right).

then introducing an utmost case of guidance by directly imposing cross-layer state sharing in both the multi-head attention and FFN blocks. With cross-layer guidance and regularization, we adapt existing Transformer models to build deep Crossformer models.

As shown in Figure 1(a), a vanilla Transformer (Vaswani et al., 2017) incorporates a multi-head attention block, a fusion layer, and an FFN block, in which the multi-head attention block uses a query-key-value decomposition to model the relationships between sequence tokens, the fusion layer combines the output of multiple heads, and the FFN block is applied to learn wider representations. Specifically, multi-head attention obtains query $\boldsymbol{Q}$, key $\boldsymbol{K}$, and value $\boldsymbol{V}$ by applying to the input three different projections, each of which consists of h linear layers (or heads) that map the d_m-dimensional input to a d_h-dimensional space, where $d_h = d_m/h$ is the head dimension. The fusion layer uses a linear layer to project the output of multiple heads to a latent feature. The FFN consists of two linear layers, where the first expands the dimension from d_m to d_f while the second reduces it from d_f to d_m. In general, Transformer-based models sequentially stack Transformer blocks to increase the network depth and hence capacity. Note that the number of parameters quickly increases as the model gets deeper. As shown in Figure 1, each block contains $12d_m^2$ parameters, in which multi-head attention contains $3d_m^2$ parameters, fusion layer d_m^2, and FFN $8d_m^2$.

2.1 CROSSFORMER-SG: A CROSS-LAYER SOFT GUIDANCE BASED REGULARIZATION

Since deep neural networks are prone to over-fitting, regularization methods are usually adopted during training to reduce the generalization error of the model (Wu et al., 2021). Different from conventional regularization methods that are often imposed on the parameters or hidden units within each layer, the proposed inductive bias of guiding lower layers with higher layers can be considered as a particular type of cross-layer structural regularization, which, according to the best of our knowledge, has not been well investigated. Specifically, rethinking the Transformer structure from this structural regularization point of view, we propose a cross-layer soft guidance not only between the keys of the multi-head attention block in the current layer and the queries of the layer above, but also between the weight matrices of the FNN blocks of two adjacent layers.

Given training data $\boldsymbol{D} = \{(x_n, y_n)\}_{n=1}^{N}$, where x_n and y_n denote the input sequence and output, respectively, the basic learning objective is to minimize the negative log-likelihood $\boldsymbol{L}_{nll}$. For a T-layer Transformer, we match the cross-layer components by minimizing the regularization loss as

$$\boldsymbol{L}_{reg} = \sum\nolimits_{t=1}^{T-1} L_{n\text{-}norm}\left(\boldsymbol{A}^{(t)}, sg(\boldsymbol{B}^{(t+1)})\right), \tag{1}$$

where $\boldsymbol{A}^{(t)}$ and $\boldsymbol{B}^{(t+1)}$ represents the states of layers t and $t+1$, respectively, and sg represents stop gradient that expresses the inductive bias of guiding lower layers with higher layers. Empirically, we denote the $\boldsymbol{A}$ and $\boldsymbol{B}$ as key ($\boldsymbol{K}$) and query ($\boldsymbol{Q}$), respectively, in self attention. Denoting α as the weight for cross-layer guidance, the training objective of Crossformer-SG can be expressed as

$$\boldsymbol{L} = \boldsymbol{L}_{nll} + \alpha \cdot \boldsymbol{L}_{reg}. \tag{2}$$

2.2 Crossformer-HG: A cross-layer hard guidance based regularization

Instead of imposing a soft guidance via the use of Equation 1, we further consider a type of hard guidance that directly lets $\boldsymbol{A}^{(t)} = \boldsymbol{B}^{(t+1)}$, which can be considered as an utmost way of injecting a cross-layer induction bias that also helps reduce the number of parameters. Below we show how to specifically adapt the operation of cross-layer hard guidance to three common blocks of Transformer.

Self-attention block. Assume there is an input sequence with l tokens, each of which has a feature of dimension d_m. For the t-th layer, the input can be defined as $\boldsymbol{H}^{(t)} \in \mathbb{R}^{l \times d_m}$, which is first projected by three separate linear layers to produce d_m-dimensional queries ($\boldsymbol{Q}^{(t)}$), keys ($\boldsymbol{K}^{(t)}$), and values ($\boldsymbol{V}^{(t)}$). As shown in Figure 1(b), to realize a hard guidance in the self-attention block, the queries in each layer are alternatively converted to the keys of the layer below. The contextual relationships between these n tokens using the scaled dot-product attention can be defined as:

$$\boldsymbol{K}^{(t)} = \begin{cases} \boldsymbol{H}^{(t)}\boldsymbol{W}_K^{(t)}, & t = T \\ \boldsymbol{Q}^{(t+1)}, & \text{Otherwise} \end{cases}, \quad \boldsymbol{Q}^{(t)} = \boldsymbol{H}^{(t)}\boldsymbol{W}_Q^{(t)}, \quad \boldsymbol{V}^{(t)} = \boldsymbol{H}^{(t)}\boldsymbol{W}_V^{(t)}, \tag{3}$$

$$\text{Att}\left(\boldsymbol{K}^{(t)}, \boldsymbol{Q}^{(t)}, \boldsymbol{V}^{(t)}\right) = \text{softmax}\left(\frac{\boldsymbol{Q}^{(t)}{\boldsymbol{K}^{(t)}}^T}{\sqrt{d_m}}\right)\boldsymbol{V}^{(t)}, \tag{4}$$

where $\boldsymbol{W}_Q^{(t)}$, $\boldsymbol{W}_K^{(t)}$, $\boldsymbol{W}_V^{(t)} \in \mathbb{R}^{d_m \times d_m}$ are learnable parameters and the softmax operation is performed row wise. Repeating this process, the cross-layer coupling of the keys and queries can be realized across all adjacent layers. This guided state sharing in the self-attention blocks can not only achieve cross-layer structural regularization, but also improve memory efficiency.

FFN block. To increase the expressiveness and capacity of Transformers, the output feature from the fusion layer in the Transformer block is delivered to the FFN block for learning wider representations. An FFN block consists of two linear layers, in which the first layer $\boldsymbol{W}_1^{(t)} \in \mathbb{R}^{d_m \times d_f}$ expands the dimension of the input from d_m to d_f, while the second layer $\boldsymbol{W}_2^{(t)} \in \mathbb{R}^{d_f \times d_m}$ reduces the dimension from d_f to d_m. Specifically, for the t-th Transformer layer, the hard-guided FFN can be defined as

$$\text{FFN}(\boldsymbol{H}^{(t)}) = \phi\left(\boldsymbol{H}^{(t)}\boldsymbol{W}_1^{(t)}\right)\boldsymbol{W}_2^{(t)}, \quad \begin{cases} \boldsymbol{W}_1^{(t)} = \boldsymbol{W}_1^{(t+1)} & t = 2, 4, ..., T-2 \\ \boldsymbol{W}_2^{(t)} = \boldsymbol{W}_2^{(t+1)} & t = 1, 3, ..., T-1 \end{cases}, \tag{5}$$

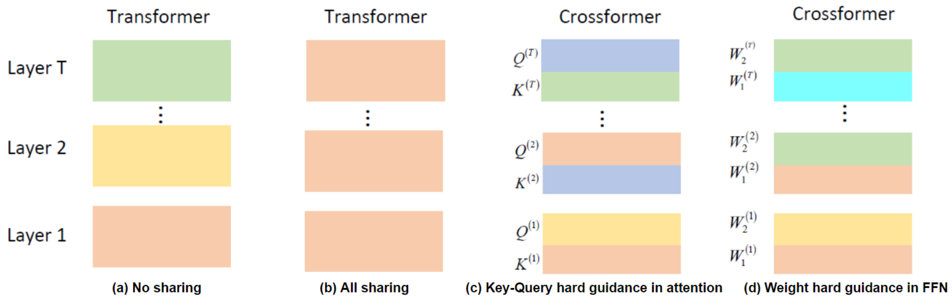
where $\boldsymbol{W}_1^{(t)} \in \mathbb{R}^{d_m \times d_f}$, $\boldsymbol{W}_2^{(t)} \in \mathbb{R}^{d_f \times d_m}$ are learnable weights. Note that this FFN block with cross-layer hard guidance can effectively reduce the number of parameters from $8d_m^2$ to $4d_m^2$.

Value and fusion layer block. As shown in Figure 1, the output feature of the self-attention block is fed into a fusion layer, which projects the output of multi-head attention as a latent feature. Thus the output of the self-attention and fusion layer blocks can be defined as:

$$\begin{aligned} &\boldsymbol{A}^{(t)} = \text{softmax}\left(\frac{\boldsymbol{Q}^{(t)}{\boldsymbol{K}^{(t)}}^T}{\sqrt{d_m}}\right), \quad \begin{cases} \boldsymbol{W}_V^{(t)} = \boldsymbol{W}_V^{(t+1)} & t = 2, 4, ..., T-2 \\ \boldsymbol{W}_F^{(t)} = \boldsymbol{W}_F^{(t+1)} & t = 1, 3, ..., T-1 \end{cases}, \\ &\text{Fus}(\boldsymbol{H}^{(t)}) = \boldsymbol{A}^{(t)}\boldsymbol{V}^{(t)}\boldsymbol{W}_F^{(t)} = \boldsymbol{A}^{(t)}\left(\boldsymbol{H}^{(t)}\boldsymbol{W}_V^{(t)}\right)\boldsymbol{W}_F^{(t)}, \end{aligned} \tag{6}$$

where $\boldsymbol{W}_V^{(t)}$, $\boldsymbol{W}_F^{(t)} \in \mathbb{R}^{d_m \times d_m}$ are learnable weights and $\boldsymbol{A}^{(t)}$ is the attention weight. Repeating this, the weights in hierarchical value and fusion layers can be alternatively shared across all layers.

Putting it all together. Given the input tokens $(x_1, ..., x_n)$, Crossformer first maps the input token to n hidden features H by embedding method. Specifically, assuming that the encoder consists of T layers, each of which comprises three sub-components: a hard-guided self-attention block, a hard-guided value and fusion layer block, and a hard-guided feed-forward block. Pre-layer normalization



(a) No sharing (b) All sharing (c) Key-Query hard guidance in attention (d) Weight hard guidance in FFN

Figure 2: Comparison of different cross-layer guidance schemes. Crossformer combines (c) and (d) but decomposes parameters in each layer into private and public parts so only the latter are shared across Crossformer.

is firstly applied to the input of each subcomponent, and then a residual skip connection (He et al., 2016) is added to its output. Formally,

$$\boldsymbol{H}_e^{(t)} = \text{FFN}\left(\text{LN}\left(\boldsymbol{O}_e^{(t)}\right); \boldsymbol{\theta}_e^{(t)}\right) + \boldsymbol{O}_e^{(t)},\ \boldsymbol{O}_e^{(t)} = \text{Att}\left(\boldsymbol{Q}_e^{(t)}, \boldsymbol{K}_e^{(t)}, \boldsymbol{V}_e^{(t)}; \boldsymbol{\theta}_e^{(t)}\right) + \boldsymbol{H}_e^{(t-1)}, \quad (7)$$

where LN $(\cdot)$, Att $(\cdot)$, FFN $(\cdot)$, and $\boldsymbol{\theta}_e^{(t)}$ are layer normalization, attention mechanism, feed-forward networks with ReLU activation, and model parameter at the t-th encoder layer, respectively. The decoder takes a similar structure as the encoder except that it includes a cross attention mechanism after each self-attention network as

$$\begin{aligned}
\boldsymbol{O}_d^{(t)} &= \text{Att}\left(\boldsymbol{Q}_d^{(t)}, \boldsymbol{K}_d^{(t)}, \boldsymbol{V}_d^{(t)}; \boldsymbol{\theta}_d^{(t)}\right) + \boldsymbol{H}_d^{(t-1)}, \\
\boldsymbol{S}_d^{(t)} &= \text{Att}\left(\text{LN}\left(\boldsymbol{O}_d^{(t)}\right), \boldsymbol{K}_e^{(T)}, \boldsymbol{V}_e^{(T)}; \boldsymbol{\theta}_d^{(t)}\right) + \boldsymbol{O}_d^{(t)}, \\
\boldsymbol{H}_d^{(t)} &= \text{FFN}\left(\text{LN}\left(\boldsymbol{S}_d^{(t)}\right); \boldsymbol{\theta}_d^{(t)}\right) + \boldsymbol{S}_d^{(t)},
\end{aligned} \quad (8)$$

where $\boldsymbol{Q}_d^{(t)}, \boldsymbol{K}_d^{(t)}, \boldsymbol{V}_d^{(t)}$ are transformed from the normalized $(t-1)$-th decoder layer, and $\boldsymbol{K}_e^{(T)}$ and $\boldsymbol{V}_e^{(T)}$ are the output of the encoder. The last layer of the decoder is used to generate the final output sequence. Note that apart from the hard-guided structure between query and key in the self-attention block and the weight matrix in the feed-forward block, we also share the weight matrix within the value and fusion layers. Thus, the Crossformer stacks $2T$ layers, each of which mainly comprises three sub-components: (1) three branches of queries, keys, and values, (2) value and fusion layers, and (3) FFN. All these layers are stacked on top of each other to build a deep neural network (see Figure 2). Note that the Crossformer can be easily adapted to not only the encoder-decoder design but also any structure including the self-attention, feed forward, or value and fusion layers.

Model proprieties. With a novel and effective cross-layer parameter sharing structure, Crossformer exhibits two attractive properties. **(1) Parameter efficient.** An effective way to enhance the representation is by increasing the number of layers (He et al., 2016). Wang et al. (2019) have studied the depth of Transformer-based networks with the number of Transformer blocks. However, the number of parameters will increase linearly as the model gets deeper. For example, the network parameters for one transformer block is $12d_m^2$. After replacing with hard-guided self-attention block, feed-forward block, and value and fusion block, it will save $1d_m^2$, $4d_m^2$, and $1d_m^2$, respectively. By using all three hard-guided blocks together, as shown in Figure 1, Crossformer will save $6d_m^2$ in total for one Crossformer block. Overall, we present a different perspective to learn deeper models by cross-layers blocks. The proposed model is friendly to the deep model which reduces the number of network parameters. **(2) Generability and scalability.** The proposed hard-guided blocks have generic architectures so that any existing attention models and multi-layer perception (MLP) models can be converted to Crossformer structure while maintaining the inherent advantages of conventional attention and MLP, such as efficiency and being simple to optimize. We further demonstrate the stability of Crossformer on some very deep models, validating the cross-layer hard guidance as a complementary structure to many deep and large Transformer model designs.

3 Experiments

Our method can be directly deployed wherever the Transformer is utilized. To test its effectiveness and general applicability, we apply our method to a diverse set of tasks, including neural machine translation, visual question answering, and graph node classification. For neural machine translation, we further study the model's generability and stability with very deep models. In the following, we present the main experimental settings and results, with more details provided in Appendix A.

3.1 NEURAL MACHINE TRANSLATION

The attention-based Transformer models have become the de-facto standards for neural machine translation tasks. To show the effectiveness and general applicability of the proposed methods, we evaluate the proposed approach on two language pairs—En-Fr and En-De—with two corpora of varying sizes, which are IWSLT (De-En) and WMT (En-De, En-Fr), while using state-of-the-art translation systems based on Transformer (Vaswani et al., 2017).

Datasets and evaluation. We benchmark the proposed models on three datasets, including (1) IWSLT'14 German-English (IWSLT De-En), (2) WMT'14 English-German (WMT'14 En-De), and (3) WMT'14 English-French (WMT'14 En-Fr). The dataset for IWSLT'14 De-En is the same as in Ranzato et al. (2015), with $160K$ sentence pairs for training, $7K$ sentence pairs for validation, and $7K$ sentence pairs for testing. For the WMT'14 En-De dataset, we follow the training corpus of Vaswani et al. (2017), which consists of about 4.5 million sentence pairs. For the WMT'14 En-Fr dataset, we replicate the setup of Gehring et al. (2017), which uses $36M$, $27K$, and $3K$ sentence pairs for training, validation, and testing respectively.

Experimental settings. We follow the encoder-decoder architecture of the Transformer-Base model (Vaswani et al., 2017) and adopt its hyper-parameters. We implement our models using Fairseq (Ott et al., 2019) and use their provided scripts for data pre-processing, training, and evaluation. For IWSLT'14 De-En, we follow the setup of Wu et al. (2019) and train all our models for $50K$ iterations with a batch size of 4K tokens. For WMT'14 En-De and WMT'14 En-Fr, we follow the training set-up of Wu et al. (2019) and train the model for $100K$ and $60K$ iterations, respectively. We use Adam (Kingma & Ba, 2014) to minimize the cross-entropy loss with a label smoothing value of 0.1 during training. For a fair comparison, we train baseline Transformer models using the same training set-up. The full details are deferred to Appendix A.

Variants of Crossformer. In order to verify the performance of each block and proposed model, we implement our model with different variants. Note that all the variants have the same model structure and training setting. The variants include (1) *Soft guidance via regularization,* regularizing the key-query in self-attention block, weight matrix in FFN block by $L2$ norm; (2) *Crossformer - single hard-guided block,* replacing the self-attention block, value-fusion block, FFN block with hard-guided structure, and named as Crossformer Key-Query, Crossformer Value, Crossformer FFN; (3) *Crossformer - All hard-guided blocks,* using the all three hard-guided blocks to replace the corresponding components in Transformer.

3.1.1 RESULTS

Results on IWSLT. On the left side of Table 1, we present the results of Crossformer-SG, which introduces soft guidance on FFN, Value, or Key-Query. BLEU scores are computed with sacrebleu which allows for a safer token-agnostic evaluation (Post, 2018). Crossformer-SG outperforms the base models in all settings, supporting our motivation of cross-layer regularization. Then on the right side of Table 1 we present the results of Crossformer-HG (hard-guide), referred to as Crossformer henceforth for brevity. While using fewer parameters, Crossformer consistently shows better performance than its Transformer counterpart. Comparing Crossformer with soft guidance with hard guidance, we find that Crossformer-SG generally delivers slightly better performance, supporting the use of hard guidance in practice given its non-trivial reduction in the number of model parameters. Overall, the results verify the merits of adding cross-layer guidance into Transformer.

Results on WMT. Experiments are conducted on standard WMT'14 English-German (DE) and English-French (FR) benchmarks. We first evaluate the 6 layers settings. As shown in Table 2, Crossformer achieves stronger performance in the translation quality under all settings. We further present deeper Crossformer-All, which includes hard-guided self attention, FFN, and value-fusion blocks. With a slightly smaller number of parameters as the 6-layer Transformer-Base model,

Table 1: BLEU scores of IWSLT translation tasks. On the left, a demonstration of the soft guidance via the regularization. On the right, we present the results of Crossformer-Hard Guidance as an utmost regularization cross-layers.

	IWSLT'14 De-En Soft Guidance		**IWSLT'14 De-En Hard Guidance**	
Model	**#Params**	**BLEU**	**#Params**	**BLEU**
Transformer	52M	33.64	52M	33.64
Crossformer-FFN	52M	34.27	39M	34.13
Crossformer-Value	52M	34.18	48M	33.94
Crossformer-Key-Query	52M	**34.51**	48M	**34.30**

Crossformer-All improves the performance by a clear margin. These results verify our conjecture that the Crossformer not only is parameter efficient but also improves the performance.

Table 2: Test results on WMT'14 benchmarks, in terms of BLEU. We further show the comparison of Crossformer All where Crossformer incorporates all three hard-guided blocks and increases the number of layers to have the same number of parameter as the Transformer.

		WMT'14 En-De		**WMT'14 En-Fr**	
Model	**Layer**	**#Params**	**BLEU**	**#Params**	**BLEU**
Transformer	6	61M	27.30	89M	38.90
Crossformer-FFN	6	47M	27.44	75M	39.37
Crossformer-Value	6	57M	27.43	85M	39.29
Crossformer-Key-Query	6	57M	**27.60**	85M	**39.55**
Crossformer-All	16	60M	**27.90**	88M	**39.63**

Table 3: The result of Crossformer and Transformer with 36 encoder layers and 6 decoder layers on WMT'14.

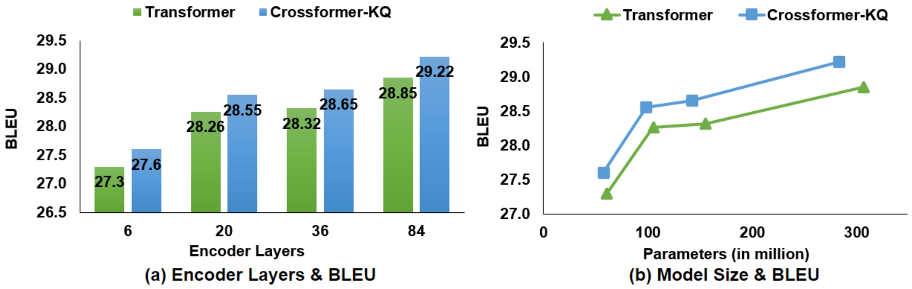
		WMT'14 En-De		**WMT'14 En-Fr**	
Model	**Layer**	**#Params**	**BLEU**	**#Params**	**BLEU**
Transformer	36	155M	28.32	183M	41.79
Crossformer-FFN	36	111M	28.47	139M	42.16
Crossformer-Value	36	142M	28.45	171M	42.15
Crossformer-Key-Query	36	142M	**28.65**	171M	**42.20**

Scaling up Crossformer. It is known that Transformer models often suffer from stability issues when the number of stacked layers grows up to a very large number (Liu et al., 2020). To verify the stability and generability of Crossformer, we conduct experiments on very deep models on both the WMT'14 En-De and En-Fr datasets (Bapna et al., 2018; Wang et al., 2019). We compare Crossformer to a Transformer-Base with a 36-layer encoder and a 6-layer decoder. The results in terms of BLEU are reported in Table 3. We observe that with fewer parameters, the 36L-6L Crossformer-Key-Query successfully trains, clearly improving the BLEU scores of the baseline on WMT'14 En-De and En-Fr by 0.33 and 0.41, respectively. The improvements are consistent with all other Crossformer settings. The relevantly large gains on the scaling up settings indicate that Crossformer has a better generalization ability in learning deeper representations.

Results on very deep models. We further train a very deep Crossformer with 84 encoder layers and 6 decoder layers on WMT'14 En-De. Figure 3 shows the performance of a Crossformer model, which improves with the increase of the network depth and the increase of the number of parameters, consistently outperform its Transformer counterpart. This shows that Crossformer is stable to train and provides attractive complementary structure to Transformer. We hypothesize that Crossformer enables better gradient propagation cross layers to improve the performance of a deep model. We leave the rigorous validation of this hypothesis as an exciting avenue for future work.

3.2 Visual question answering

We consider a multi-modal learning task, visual question answering (VQA) (Goyal et al., 2017), where the model requires to have a fine-grained and simultaneous understanding of both the visual content of images and textual content of questions. Modular Co-Attention Network (MCAN) (Yu et al., 2019) has been proposed to learn the image-question relationship to correctly answer questions.



(a) Encoder Layers & BLEU

(b) Model Size & BLEU

Figure 3: **Scaling up Crossformer.** The performance of Crossformer improves with an increase in (1) the number of encoder layers and (2) the number of parameters on the WMT'14 En-De corpus. The Crossformer consistently outperform the Transformer indicating the generability and scalibility of Crossformer and the complementary of Crossformer to different model designs.

Table 4: Accuracies and PAvPUs of different attentions on both the original VQA-v2 dataset and the noise ones.

	Accuracy ↑		PAvPU ↑	
	Original	Noisy	Original	Noisy
Base	66.74	63.58	71.96	**68.29**
Crossformer-FFN	66.89 ±0.01	64.16 ±0.02	71.95 ±0.04	68.23 ±0.03
Crossformer-Key-Query	**66.92** ±0.02	**64.23** ±0.01	**72.03** ±0.03	68.27 ±0.02

We adapt the proposed Crossformer structure to MCAN and compare it with the original model structure. We conduct experiments on the VQA-v2 dataset (Goyal et al., 2017) and follow the hyperparameters and other settings from Yu et al. (2019). In addition, to test the robustness of Crossformer, we perturb the input by incorporating the Gaussian noise (mean 0, variance 1) to image features (Larochelle et al., 2007; Fan et al., 2020). We use four-layer encoder-decoder based MCAN as the baseline model. The accuracy is reported for both the original data and noisy data. As in Fan et al. (2020), we use the hypothesis testing based Patch Accuracy vs Patch Uncertainty (PAvPU) (Fan et al., 2020; Mukhoti & Gal, 2018) as a measure of the uncertainty estimation where the p-value threshold is set to be 0.05 and the number of attention weight samples is 20. Please see detailed experimental settings in Appendix A.

Results. In Table 4, we report the accuracy and uncertainty of different Transformer structures on both original and noisy data. For accuracy, it shows that Crossformer consistently outperforms the based model on both original and noisy data. For uncertainty, we observe that Crossformer has on par uncertainty estimations on both original and noisy data. These results show that Crossformer is more robust to the noise which demonstrate the better layer interaction with this hard-guided structure.

3.3 GRAPH NODE CLASSIFICATION

To demonstrate the general applicability of Crossformer, we also experiment the method with graph attention networks (GAT) (Veličković et al., 2017). The graph structure is incorporated into the attention masks in which nodes are able to attend to their neighborhoods' features. Relying on the self-attention layers, GAT processes node-features for graph node classification.

Experimental Setup. We conduct experiments on three benchmark graphs including - Cora, Citeseer, and Pubmed (Sen et al., 2008) following the setting in GAT (Veličković et al., 2017). All experiments are conducted in the transductive setting, where all nodes from training and test are on the same graph (Yang et al., 2016). The details of three datasets and experimental settings are deferred to Appendix A.

Results. In Table 5, we report the mean classification accuracies on the test nodes over 5 random runs and the standard deviations of Crossformer. We experiment with Crossformer-Key-Query and Crossformer-FFN. Table 5 shows that hard guidance consistently improves upon the corresponding baseline models across all three datasets, which further confirms the efficient structure of this cross-layer structure with less computational memory. In addition, the key-query guided structure performs better than the FFN guided structure, which agrees with previous observations.

Table 5: Classification accuracy for graphs.

Attention	Cora	Citeseer	PubMed
GAT	83.00	72.50	77.26
Crossformer-FFN	83.42 ±0.2	73.11 ±0.2	77.72 ±0.3
Crossformer-Key-Query	**83.81** ±0.3	**73.40** ±0.2	**77.95** ±0.2

3.4 Ablation Study

We conduct ablation study with Crossformer to exam the role of the alternated cross-layer hard guidance by sharing specific components across all adjacent layers. We find that the experimental results are not sensitive to the choice of the hard guidance and the order of hard guidance. Any guidance structures would give similar results and outperform the baseline model in all settings. In all experiments considered in the paper, which cover various tasks and model sizes, we have simply fixed it as key-query and w_2-w_1. Please see detailed results in Table 6 in the Appendix.

4 Conclusion and Discussion

We propose Crossformer that introduces alternated cross-layer guidance into Transformer, providing a novel type of cross-layer structural regularization that not only reduces the number of parameters but also improves the model performance. The proposed cross-layer guidance, which can be injected into standard Transformer blocks, including the multi-attention head, fusion, and feed-forward network blocks, requires surprisingly few modifications to the standard model and hence enables us to easily convert existing Transformer-based models to Crossformer-based ones. Our experiments on a variety of neural machine translation tasks show that Crossformer achieves strong performance in accuracy with less computational memory. Further, on visual question answering, graph node classification, and very deep models, Crossformer demonstrates its general applicability and stability, showing its great potential to become a standard alternative to many existing Transformer models.

In addition to the proposed Crossformer, several methods have been introduced to improve and understand the parameter efficiency in Transformer. The first line of research is focused on simpler attention mechanism (Kovaleva et al., 2019; Chelba et al., 2020; Katharopoulos et al., 2020; Liu et al., 2021). The second line of research is focused on improving efficiency by sharing parameters across layers in deep neural networks (Lan et al., 2019; Lee et al., 2020). Lan et al. (2019) demonstrate that cross-layer parameter sharing in Transformer leads to a lighter and faster-to-train model without sacrificing the performance on various language understanding benchmarks. Lee et al. (2020) show the input token distributions may each exhibit different dynamics, yet together share certain regularities because they all come from the same data. The third line of research is focused on understanding the capabilities and function of different Transformer components (Tenney et al., 2019a; Liu et al., 2019; Tenney et al., 2019b; Phang et al., 2021). We consider cross-layer hard guidance as a parameter efficient method to leverage the existing efficient Transformer architecture to build the entire Crossformer network.

In addition to the proposed Crossformer, there is a rich set of recent works on improving the Transformer architecture, including: (1) learning a better representation — for example, using convolutions to improve the expressiveness (Wu et al., 2019), incorporating gated units (Dauphin et al., 2017), or utilizing multi-branch feature extractors (So et al., 2019); (2) interpreting the multi-head attention — for example, synthetic attention matrices (Tay et al., 2021) improving performances and more Transformer heads leading to redundant representations (Michel et al., 2019); and (3) improving efficiency — for example, using compression (Sun et al., 2020), pruning (Voita et al., 2019), and distillation (Sanh et al., 2019). The proposed Crossformer leverages the existing Transformer designs and components and, to the best of our knowledge, is the first to propose cross-layer soft/hard guidance as a structural regularization. This general and efficient framework gives us the flexibility to better utilize the information of different layers and facilitate their interaction. It would be interesting to investigate whether Crossformer can be synergized with other structural modifications proposed for Transformer, which we leave as promising research topic for future study.

REPRODUCIBILITY STATEMENT

While we show improvements brought by our work on a variety of tasks from a broad range of domains, our framework is general enough that it could be used to improve potentially any Transformer based models. Training a state-of-the-art Transformer model now requires substantial computational resources which demands considerable energy, along with the associated financial and environmental costs. Our proposed Crossformer aims to reduce the computational memory making it accessible for researchers with limited computations and further ensuring the reproducibility. The detailed descriptions of the datasets and experimental settings are included in the main paper as well as the Appendix. We will release the code after the anonymity period.

REFERENCES

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Ankur Bapna, Mia Xu Chen, Orhan Firat, Yuan Cao, and Yonghui Wu. Training deeper neural machine translation models with transparent attention. *arXiv preprint arXiv:1808.07561*, 2018.

Ciprian Chelba, Mia Chen, Ankur Bapna, and Noam Shazeer. Faster transformer decoding: N-gram masked self-attention. *arXiv preprint arXiv:2001.04589*, 2020.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Marcella Cornia, Matteo Stefanini, Lorenzo Baraldi, and Rita Cucchiara. Meshed-memory Transformer for image captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10578–10587, 2020.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pp. 933–941. PMLR, 2017.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Yuntian Deng, Yoon Kim, Justin Chiu, Demi Guo, and Alexander Rush. Latent alignment and variational attention. In *Advances in Neural Information Processing Systems*, pp. 9712–9724, 2018.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*, 2018.

Xinjie Fan, Shujian Zhang, Bo Chen, and Mingyuan Zhou. Bayesian attention modules. *Advances in Neural Information Processing Systems*, 33, 2020.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. Convolutional sequence to sequence learning. In *International Conference on Machine Learning*, pp. 1243–1252. PMLR, 2017.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6904–6913, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pp. 5156–5165. PMLR, 2020.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*, 2019.

Anders Krogh and John A Hertz. A simple weight decay can improve generalization. In *Advances in neural information processing systems*, pp. 950–957, 1992.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.

Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *Proceedings of the 24th international conference on Machine learning*, pp. 473–480, 2007.

Sangho Lee, Youngjae Yu, Gunhee Kim, Thomas Breuel, Jan Kautz, and Yale Song. Parameter efficient multimodal transformers for video representation learning. *arXiv preprint arXiv:2012.04124*, 2020.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European conference on computer vision*, pp. 740–755. Springer, 2014.

Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021.

Nelson F Liu, Matt Gardner, Yonatan Belinkov, Matthew E Peters, and Noah A Smith. Linguistic knowledge and transferability of contextual representations. *arXiv preprint arXiv:1903.08855*, 2019.

Xiaodong Liu, Kevin Duh, Liyuan Liu, and Jianfeng Gao. Very deep transformers for neural machine translation. *arXiv preprint arXiv:2008.07772*, 2020.

Yiping Lu, Zhuohan Li, Di He, Zhiqing Sun, Bin Dong, Tao Qin, Liwei Wang, and Tie-Yan Liu. Understanding and improving transformer from a multi-particle dynamic system point of view. *arXiv preprint arXiv:1906.02762*, 2019.

Paul Michel, Omer Levy, and Graham Neubig. Are sixteen heads really better than one? *arXiv preprint arXiv:1905.10650*, 2019.

Jishnu Mukhoti and Yarin Gal. Evaluating Bayesian deep learning methods for semantic segmentation. *arXiv preprint arXiv:1811.12709*, 2018.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*, 2019.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pp. 311–318. Association for Computational Linguistics, 2002.

Jason Phang, Haokun Liu, and Samuel R Bowman. Fine-tuned transformers show clusters of similar representations across layers. *arXiv preprint arXiv:2109.08406*, 2021.

Matt Post. A call for clarity in reporting bleu scores. *arXiv preprint arXiv:1804.08771*, 2018.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *AI magazine*, 29(3):93–93, 2008.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

David So, Quoc Le, and Chen Liang. The evolved transformer. In *International Conference on Machine Learning*, pp. 5877–5886. PMLR, 2019.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. *arXiv preprint arXiv:2004.02984*, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Yi Tay, Dara Bahri, Donald Metzler, Da-Cheng Juan, Zhe Zhao, and Che Zheng. Synthesizer: Rethinking self-attention for transformer models. In *International Conference on Machine Learning*, pp. 10183–10192. PMLR, 2021.

Damien Teney, Peter Anderson, Xiaodong He, and Anton Van Den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4223–4232, 2018.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. Bert rediscovers the classical nlp pipeline. *arXiv preprint arXiv:1905.05950*, 2019a.

Ian Tenney, Patrick Xia, Berlin Chen, Alex Wang, Adam Poliak, R Thomas McCoy, Najoung Kim, Benjamin Van Durme, Samuel R Bowman, Dipanjan Das, et al. What do you learn from context? probing for sentence structure in contextualized word representations. *arXiv preprint arXiv:1905.06316*, 2019b.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pp. 10347–10357. PMLR, 2021.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*, 2019.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.

Zhen Wu, Lijun Wu, Qi Meng, Yingce Xia, Shufang Xie, Tao Qin, Xinyu Dai, and Tie-Yan Liu. Unidrop: A simple yet effective technique to improve transformer without extra cost. *arXiv preprint arXiv:2104.04946*, 2021.

Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. Revisiting semi-supervised learning with graph embeddings. *arXiv preprint arXiv:1603.08861*, 2016.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6281–6290, 2019.

Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J Kim. Graph transformer networks. *Advances in Neural Information Processing Systems*, 32:11983–11993, 2019.

A Experimental details

A.1 Neural Machine Translation

A.1.1 Model Specifications

Following the Neural Machine Translation (NMT) setting from Vaswani et al. (2017), the basic configurations of the Transformer architecture are the base settings. Both of them consist of a 6-layer encoder and 6-layer decoder. The size of the hidden nodes and embeddings is set to 512. The number of heads is 8 for the base. For all settings, the dimensionality of the inner-layer of the position-wise FFN is four times of the dimensionality of the hidden states.

A.1.2 Experimental Settings

We use the Adam (Kingma & Ba, 2014) optimizer and follow the optimizer setting and learning rate schedule in (Vaswani et al., 2017). We employ label smoothing value of 0.1 (Szegedy et al., 2016) in all experiments. For a fair comparison, we trained baseline Transformer models using the same training set-up. We use BLEU (Papineni et al., 2002) as the evaluation measure for machine translation. During inference, we use beam search with beam size 4 and length penalty 0.6 for WMT14, and beam size 5 and length penalty 1.0 for IWSLT'14, following Vaswani et al. (2017). For the soft guidance in the IWSLT'14, we fix the guidance-weight hyperparameter α in Equation 2 as 0.01. For the WMT'14 En-De dataset, all the data are tokenized and segmented into subword symbols using jointly BPE with $32K$ merge operations. For the WMT'14 En-Fr dataset, the $40K$ vocabulary is based on a joint source and target BPE factorization.

A.2 Visual Question Answering

A.2.1 Model Specifications

We use the state-of-art VQA models, MCAN Yu et al. (2019) which consists of MCA layers. Two types of attention in the MCA layer are self-attention (SA) over questions and image features and guided-attention (GA) between question and image features. Mult-head structure is included in each MCA layer with the residual and layer normalization components. By stacking multiple MCA layers, MCAN gradually extracts the image and question features through the encoder-decoder structure. Four co-attention layers' MCAN is used in our experiment.

A.2.2 Experimental Settings

We conduct experiments on the VQA-v2 dataset Goyal et al. (2017), consisting of human-annotated question-answer pairs for images from the MS-COCO dataset Lin et al. (2014). The whole dataset is split into three parts. For training, there are 40k images and 444k QA pairs. For validation, there are 40k images and 214k QA pairs. For testing, there are 80k images and 448k QA pairs. The evaluation is conducted on the validation set as the true labels for the test set are not publicly available (Deng et al., 2018). For the noisy dataset, we perturb the input by adding Gaussian noise (mean 0, variance 1) to the image features Larochelle et al. (2007). We use the same model hyperparameters and training settings in Yu et al. (2019) as follows: the dimensionality of input image features, input question features, and fused multi-modal features are set to be 2048, 512, and 1024, respectively. The latent dimensionality in the multi-head attention is 512, the number of heads is set to 8, and the latent dimensionality for each head is 64. The size of the answer vocabulary is set to $N = 3129$ using the strategy in Teney et al. (2018). To train the MCAN model, we use the Adam optimizer (Kingma & Ba, 2014) with $\beta_1 = 0.9$ and $\beta_2 = 0.98$. The base learning rate is set to $\min(2.5te^{-5}, 1e^{-4})$, where t is the current epoch number starting from 1. After 10 epochs, the learning rate is decayed by $1/5$ every 2 epochs. All the models are trained up to 13 epochs with the same batch size of 64.

A.3 Graph Neural Networks

A.3.1 Model Specifications

Following the setting in Veličković et al. (2017), we use the two-layer GAT model. Glorot initialization Glorot & Bengio (2010) is utilized. The initial learning rate is 0.01 for Pubmed and 0.005 for all other datasets. The model is trained with the cross-entropy loss using the Adam SGD optimizer Kingma & Ba (2014).

A.3.2 Experimental Settings

We follow the architecture and hyperparameters settings in (Veličković et al., 2017). The number of attention heads is 8 in the first layer followed by an exponential linear unit (ELU) Clevert et al. (2015) nonlinearity and is 1 in the second layer for classification. Dropout Srivastava et al. (2014) is set as $p = 0.6$. On Cora and Citeseer, we apply $L2$ regularization with $\lambda = 0.0005$ during training. Pubmed required slight changes to the architecture. The second layer has 8 attention heads and the weight λ of $L2$ regularization is 0.001. We adopt the early stopping strategy on both the cross-entropy loss and accuracy on the validation nodes Sen et al. (2008). The patience is 100 epochs.

A.4 Ablation Study

Table 6: Ablation study of Crossformer alternatively hard guidance on IWSLT'14 De-En.

Crossformer Blocks	Guidance Type	Bleu
Self Attention	Query Query	34.05
	Key Key	34.07
	Query Key	34.18
	Key Query	**34.30**
FFN	w_1 w_2	33.82
	w_2 w_1	**34.13**