Scheduled DropHead: A Regularization Method for Transformer Models

Wangchunshu Zhou[1] *   **Tao Ge**[2]   **Ke Xu**[1]   **Furu Wei**[2]   **Ming Zhou**[2]

[1]Beihang University, Beijing, China

[2]Microsoft Research Asia, Beijing, China

`zhouwangchunshu@buaa.edu.cn, kexu@nlsde.buaa.edu.cn`
`{tage, fuwei, mingzhou}@microsoft.com`

Abstract

We introduce DropHead, a structured dropout method specifically designed for regularizing the multi-head attention mechanism which is a key component of transformer. In contrast to the conventional dropout mechanism which randomly drops units or connections, Drop-Head drops entire attention heads during training to prevent the multi-head attention model from being dominated by a small portion of attention heads. It can help reduce the risk of overfitting and allow the models to better benefit from the multi-head attention. Given the interaction between multi-headedness and training dynamics, we further propose a novel dropout rate scheduler to adjust the dropout rate of DropHead throughout training, which results in a better regularization effect. Experimental results demonstrate that our proposed approach can improve transformer models by 0.9 BLEU score on WMT14 En-De translation task and around 1.0 accuracy for various text classification tasks.

1 Introduction

The transformer (Vaswani et al., 2017) architecture has shown state-of-the-art performance across a variety of Natural Language Processing (NLP) tasks. One key architectural innovation of the transformer model is the multi-head attention mechanism (MHA) where attention is computed independently by multiple parallel attention heads. Intuitively, the MHA mechanism allows the model to jointly attend to information from different representation subspaces at different positions. In practice, however, some recent studies (Voita et al., 2019; Michel et al., 2019) show that in most cases, the MHA mechanism is dominated by only a small portion of attention heads which

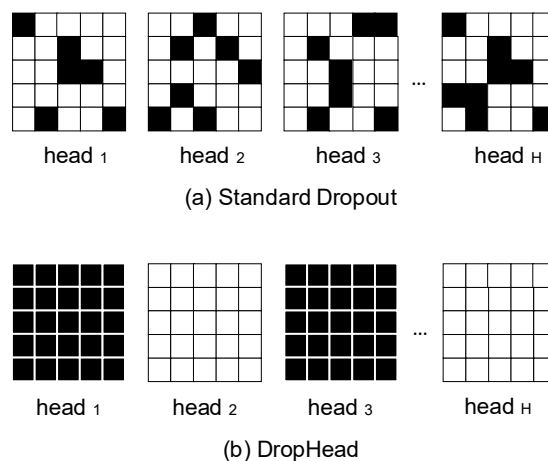*This work was done during the first author's internship at Microsoft Research Asia.

Figure 1: Illustration of DropHead : (a): standard Dropout randomly drops units in each attention heads. (b) DropHead drops entire attention heads.

contributes most to the output. The other attention heads can even be removed at test time without significantly impacting the performance, indicating that the MHA mechanism is far from being fully exploited.

To address the limitation, we propose Drop-Head, a structured dropout mechanism specifically designed to regularize the MHA mechanism in transformer models. Different from conventional dropout approaches (Srivastava et al., 2014) that randomly dropout activations or connections, DropHead drops an entire attention head with a pre-defined probability, which is illustrated in Figure 1. Under the DropHead mechanism, any attention head is possible to be dropped out during training, which requires every attention head to work hard so that it can ensure the model to function well even if other heads are dropped out. As a result, DropHead can prevent the MHA from being dominated by one or a few attention heads and encourage more attention heads to encode

useful information. Also, it helps MHA mechanism generalize better by breaking up excessive co-adaptation between multiple attention heads.

Moreover, considering the training dynamics (Michel et al., 2019) of the MHA where the aforementioned head domination phenomenon tends to happen at the very beginning of the training process, we propose a specific dropout rate scheduler for the DropHead mechanism, which looks like a V-shaped curve: it applies a relatively high dropout rate for the beginning and ending epochs during training to prevent the MHA mechanism from being dominated by a few attentions heads and reduce overfitting, while uses a lower dropout rate in the remaining steps to allow the model to better fit the training data.

We conduct experiments on both neural machine translation and text classification benchmark datasets. Experimental results show that the proposed Scheduled DropHead yields consistent and substantial improvements over the standard transformer model (0.9 BLEU score on WMT2014 En-De translation dataset and around 1.0 accuracy on various text classification datasets) without any other modification in the model architecture. In addition, we also find that Scheduled DropHead enables a trained transformer model to better retain its performance after pruning out a relatively large number of attention heads.

Our contribution can be summarized as follows:

- We propose DropHead, a simple and versatile regularization method for transformer models. It helps avoid the head domination problem, which enables more attention heads to encode useful information and prevents excessive co-adaptation between attention heads, improving the generalization ability of the model.

- We introduce a dropout scheduler for DropHead, which adjusts the dropout rate throughout the training process to strengthen the effect of DropHead, which further improves the performance.

2 Background

In this section, we briefly recall the important background notations including attention mechanism (Bahdanau et al., 2014), multi-head attention mechanism (Vaswani et al., 2017), and dropout (Srivastava et al., 2014).

Attention Mechanism 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. We focus on the dot-product attention (Luong et al., 2015) which is employed in the transformer architecture.

Given a sequence of vectors $H \in \mathbb{R}^{l \times d}$, where l and d represent the length of and the dimension of the input sequence, the self-attention projects H into three different matrices: the query matrix Q, the key matrix K and the value matrix vector V, and uses scaled dot-product attention to get the output representation:

$$
\begin{aligned}
Q, K, V &= HW^Q, HW^K, HW^V \\
\text{Attn}(Q, K, V) &= \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) V
\end{aligned} \tag{1}
$$

where W^Q, W^K, W^V are learnable parameters and $\text{softmax}()$ is performed row-wise.

Multi-Head Attention To enhance the ability of self-attention, multi-head self-attention is introduced as an extension of the single head self-attention, which jointly model the multiple interactions from different representation spaces:

$$
\begin{aligned}
\text{MHA}(H) &= [\text{head}_1; \ldots; \text{head}_H] W^O, \\
\text{where head}_i &= \text{Attn}\left(HW_i^Q, HW_i^K, HW_i^V\right)
\end{aligned} \tag{2}
$$

where W^O, W_i^Q, W_i^K, W^V are learnable parameters. The transformer architecture is a stack of several multi-head self-attention layers and fully-connected layers. The multi-head attention mechanism extends the standard attention mechanism (Bahdanau et al., 2014) by computing attention with H independent attention heads, thus allows the model to jointly attend to information from different representation subspaces at different positions.

Dropout Since introduced by Srivastava et al. (2014), dropout is extensively used in training deep neural networks. The method randomly suppresses neurons during training by setting it to 0 with probability p, which is called the dropout rate. The resulting model at test time is often interpreted as an average of multiple models during training, and it is argued that this can reduce

overfitting and improve test performance. Dropout achieves great success on fully-connected layers. However, vanilla dropout fails to improve the performance of convolutional neural networks or recurrent neural networks and is only applied in the feed-forward layers in the transformer model, which motivates our proposed method.

3 Scheduled DropHead

In this section, we begin by introducing the proposed DropHead mechanism in detail and discuss its potential effects on training transformer models. Afterwards, we describe the proposed dropout rate scheduler specifically designed for DropHead.

3.1 DropHead

DropHead is a simple regularization method similar to Dropout (Srivastava et al., 2014) and DropConnect (Wan et al., 2013). The main difference between DropHead and conventional Dropout is that we drop an entire attention head with a predefined probability in DropHead, instead of dropping individual units in conventional Dropout.

Formally, during training, for a transformer layer with H attention heads, we randomly sample a binary mask vector $\xi \in \{0,1\}^H$ with each element independently sampled from $\xi_i \sim$ Bernoulli(p) where p is the dropout rate. Afterward, we create H attention head masks M, which is of the same shape with the output of one attention head, by filling all its elements with the corresponding ξ_i. We then apply the mask vectors M_i to the corresponding output of attention heads head$_i$ by element-wise multiplication. Finally, we normalize the output of MHA to ensure the scale of the output representation matches between training and inference. The output of MHA mechanism equipped with DropHead can be formulated as:

$$
\begin{aligned}
\text{MHA}(H) &= ([\text{head}_1 \odot M_i; \ldots; \text{head}_H \odot M_H] \, W^O)/\gamma, \\
\text{where head}_i &= \text{Attn}\left(HW_i^Q, HW_i^K, HW_i^V\right); \\
\gamma &= \text{sum}(\xi)/H
\end{aligned}
$$
(3)

where γ is the rescale factor used for normalization and $\odot$ denotes element-wise product. The procedure of DropHead is summarized in Algorithm 1.

Effect of DropHead Applying DropHead during training forces the MHA mechanism to optimize the same objective with different randomly selected subsets of its attention heads. Its potential effects can be summarized as follows: It prevents

---

Algorithm 1 DropHead

Require: Dropout rate p; outputs from H attention heads $O = [\text{head}_1; \ldots; \text{head}_H]$; mode
1: **if** mode == inference **then**
2:   return O
3: **end if**
4: Randomly sample ξ: $\xi_i \sim$ Bernoulli(p).
5: For each element ξ_i, create a mask vector M_i of the same shape as head$_i$ with all elements equal to ξ_i.
6: Apply the mask: $O = O \times M$
7: Normalization: $O = O \times H/\mathbf{count\_ones}(\xi)$

---

the MHA from being dominated by few attention heads and reduces excessive co-adaptation between different attention heads, which helps MHA exploit its multi-headness more effectively and reduces the risk of overfitting, thus improving the generalization ability of trained models.

3.2 Dropout rate scheduler

Motivated by Curriculum Dropout (Morerio et al., 2017) and other structured dropout approaches (Ghiasi et al., 2018; Zoph et al., 2018), we propose to apply a dropout rate scheduler to adjust the effect of DropHead throughtout the training process. Different from the previous scheduled dropout approaches that generally initialize the training procedure with a dropout rate of 0 and gradually increase the dropout rate for alleviating the excessive co-adaptation in the late training steps, we propose a novel V-shaped dropout rate scheduler for DropHead, as shown in Figure 2 which additionally applies high dropout rate at the early stage of training. The motivation of high dropout rate at the beginning of training is to prevent the MHA from being dominated by few attention heads, which proves to happen at the very early stage of training, as the previous study (Michel et al., 2019) shows.

Specifically, we start the training process with a relatively high dropout rate of p_{start} and linearly decrease it to 0 during the early stage of training, which is empirically chosen to be the same training steps for learning rate warmup. Afterwards, we linearly increase the dropout rate to p_{end} in the remaining training steps following existing dropout rate schedulers in DropBlock and ScheduledDropPath. To avoid introducing additional hyperparameters, we empirically set $p_{start} = p_{end}$ and find it generally works well across different
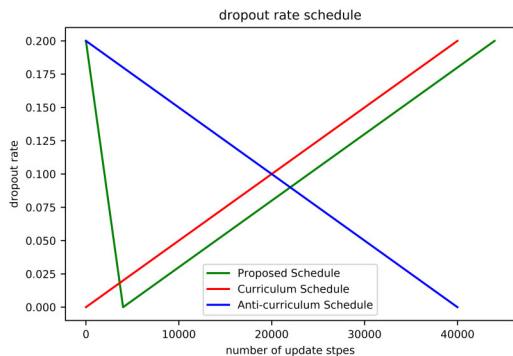
Figure 2: Illustration of the proposed dropout rate schedule, the curriculum dropout rate schedule, and the anti-curriculum dropout rate schedule.

tasks.

4 Experiments

In this section, we first conduct experiments on several benchmark datasets on machine translation and text classification to demonstrate the effectiveness of Scheduled DropHead. Afterward, we conduct experiments on pruning transformer models trained with DropHead to test if it succeeds in preventing dominating attention heads and improving attention head pruning.

4.1 Machine Translation

We first conduct experiments on the machine translation task, which is one of the primary applications of the transformer model.

Dataset Following Vaswani et al. (2017), we use the WMT 2014 English-German dataset consisting of about 4.5 million sentence pairs as the training set. The newstest2013 and newstest2014 are used as the development set and the test set. We apply BPE (Sennrich et al., 2015) with 32K merge operations to obtain subword units.

Models We train the **transformer-big** (Vaswani et al., 2017) model with the proposed Scheduled DropHead method as well as its counterpart where no dropout rate scheduler is employed. To ensure fair comparisons, we compare our models against the result reported by Vaswani et al. (2017) and our reproduced result. In addition, as our approach is expected to be able to reduce excessive co-adaptation between multiple attention heads, we also experiment with a variant of the transformer model where we increase the number of attention heads in each layer from 16 to 32. We denote the

modified transformer model as **transformer-big (more heads)**. Note that we change the number of transformer heads by reducing the hidden dimension size of each attention heads, thus do not increase the number of total parameters in the resulting model. For reference, we also compare our model against that trained by applying the vanilla dropout on self-attention layers in the transformer model, as well as several recent techniques improving the transformer model in Table 1 to futher demonstrate the effectiveness of our approach.

Training We generally follow the hyperparameter setting provided in (Vaswani et al., 2017). We set the dropout rate for DropHead to 0.2 and decrease the dropout rate in feed-forward layers from 0.3 to 0.2 based on dev set performance, which is consistent with the fact that DropHead mechanism already provides some regularization effects.

Evaluation Following previous work, we average the last 10 checkpoints for evaluation. All results use beam search with a beam width of 4 and length penalty of 0.6.

Models	BLEU	PPL
Weighted Transformer(Ahmed et al., 2017)	28.9	-
Tied Transformer(Xia et al., 2019)	29.0	-
Layer-wise Coordination(He et al., 2018)	29.1	-
Transformer-big	28.4	-
- ours (reproduced)	28.7	4.32
- + Attention dropout	28.7	4.29
- + DropHead	29.2	4.15
- + Scheduled DropHead	29.4	4.08
Transformer-big (more heads)	28.4	4.39
- + Attention dropout	28.5	4.38
- + DropHead	29.3	4.12
- + Scheduled DropHead	**29.6***	**4.02**

Table 1: Machine translation performance on WMT14 en-de and newstest2014 of compared models. The results in the first section are the results reported on the corresponding publications. Other results are reported as the median of 5 random runs. * denotes statistically significant with $p < 0.05$ compared with all baselines.

Results Table 1 shows the BLEU score and the perplexity of different compared models. We find that applying vanilla dropout on self-attention layers generally fails to improve the model's performance. In contrast, the DropHead mechanism is able to improve the transformer model by around 0.5 BLEU score and 0.2 perplexity, demonstrating its ability to improve the model's generalization ability. In addition, the proposed dropout rate

scheduler can further improve the performance of the strong baseline of Transformer + DropHead by 0.2 BLEU score, which confirms its effectiveness.

We also find that simply increasing the number of attention heads in each layer yields slightly worse result compared with the default model configuration. However, training the transformer model with more attention heads with Scheduled DropHead can yield further improvement compared with the original model architecture. This suggests that the proposed method is able to provide good regularization effect for the MHA in transformer. Our approach also substantially outperforms several existing techniques on improving the transformer model that do not involve large modification of the model architecture and additional training data, which further demonstrates its effectiveness.

4.2 Text Classification

We also conduct experiments on sentence classification tasks to further demonstrate the effectiveness of the proposed methods.

Datasets	Type	#Classes	#Documents
IMDB	Sentiment	2	50000
Yelp Review	Sentiment	2	598000
AG's News	Topic	4	127,600
DBPedia	Topic	14	630,000
TREC	Question	6	5,952
Yahoo! Answers	Question	10	1,146,000
SNLI	Inference	3	550,152

Table 2: Statistics for text classification datasets.

Datasets We conduct experiments on text classification datasets ranging from different tasks. Statistics of datasets are listed in Table 2. All datasets are split into training, development and testing sets following previous work. For sentiment analysis, we use the binary film review **IMDB** dataset (Maas et al., 2011) and the binary version of the **Yelp Review** dataset built by (Zhang et al., 2015). For topic classification task, we employ **AG's News** and **DBPedia** created by (Zhang et al., 2015). For question classification task, we evaluate our method on the six-class version of the **TREC** dataset (Voorhees and Tice, 1999) and **Yahoo! Answers** dataset created by (Zhang et al., 2015). We also evaluate on the **SNLI** dataset (Bowman et al., 2015) which is a collection of sentence pairs labeled for entailment, contradiction, and semantic independence.

Models We employ BERT-base (Devlin et al., 2018) as a strong baseline and compare the performance of fine-tuning it without attention dropout, with vanilla attention dropout, and with DropHead/Scheduled DropHead. However, the dominant attention heads in the BERT model may already be formed during pretraining, which may hinder the effect of DropHead, and pretraining BERT with our approach is beyond the scope of our paper. Therefore, we also employ a smaller, randomly initialized (i.e. without pretraining) transformer-based text classification model as our baseline model. The model has a hidden size of 768, 6 transformer blocks and 12 self-attention heads. Following (Sun et al., 2019), we train the models with a batch size of 24, a dropout rate of 0.1 for both vanilla Dropout and DropHead. We employ the Adam optimizer with a base learning rate as 2e-5 or 1e-4 with a warm-up ratio of 0.1 for the models with and without pretraining respectively. We train the models for 10 epochs and select the best model according to the accuracy on the held out dev set.

Results We report the accuracy of compared models on the test set in Table 3. We can see that applying Scheduled DropHead can significantly (with $p < 0.05$) improve the performance of vanilla transformer-based text classification models, as well as that applying vanilla dropout on the MHA mechanism of transformer models, on 6 out of 7 datasets. This demonstrates that our proposed approach works well for natural language understanding tasks. Scheduled DropHead can also consistently improve the performance of a very competitive BERT-based model with an arguably significance (with $p < 0.1$) for most datasets. This further confirms the effectiveness of our approach. The performance improvement yielded by the proposed approach is much larger when applying on the vanilla transformer-based text classification model without pretraining. We suspect this may be because the BERT model is already approaching the state-of-the-art performance on each datasets and the dominant attention heads have already been formed in the pretrained BERT model during its pretraining procedure, which reduces the improvement yielded by Scheduled DropHead. Pretraining BERT models with DropHead may alleviate this problem, we leave this for the future work. In addition, we find that the Scheduled DropHead outperforms its counterpart where

Model	IMDB	Yelp	AG	DBPedia	TREC	Yahoo!	SNLI
Char-level CNN(Zhang et al., 2015)	-	95.12	90.49	98.45	-	71.20	-
VDCNN(Conneau et al., 2017)	-	95.72	91.33	98.71	-	73.43	-
DPCNN(Johnson and Zhang, 2017)	-	97.36	93.13	99.12	-	76.10	-
ULMFiT(Howard and Ruder, 2018)	95.40	97.84	94.99	99.20	96.40	-	-
BERT-base (Devlin et al., 2018)	94.60	97.61	94.75	99.30	97.20	77.58	90.73
-Attention Dropout	94.51	97.64	94.65	99.28	97.20	77.64	90.65
-DropHead	95.22	97.77	94.90	99.35	**97.60**	**78.05**	90.85
-Scheduled DropHead	**95.48***	<u>**97.92**</u>	**94.95**	<u>**99.41**</u>	<u>**97.80**</u>	77.93	<u>**90.92**</u>
Transformer (w/o pretraining)	90.85	95.61	91.08	98.69	94.60	73.15	86.52
-Attention Dropout	90.91	95.56	91.12	98.62	94.40	73.31	86.57
-DropHead	91.87	96.14	91.68	98.85	**95.20**	74.38	87.14
-Scheduled DropHead	**91.95***	**96.21***	**91.74***	**98.91**	**95.60***	**74.41***	**87.20***

Table 3: Test set results (accuracy) of compared models on text classification benchmarks. The results in the first section are the results reported on the corresponding publications. Other results are reported as the median of 5 random runs. '−' means not reported results. * and _ means statistically significant improvement upon the corresponding baseline with $p < 0.05$ and $p < 0.1$.

a constant dropout rate is applied for DropHead. This confirms the effectiveness of the proposed dropout rate schedule.

4.3 Analysis

In this section, we perform experiments following that in (Michel et al., 2019) to analyze the relative importance of individual attention heads in transformer model trained on WMT14 en-de dataset with and without the proposed Scheduled DropHead mechanism, and try to prune attention heads to improve its efficiency.

Model	Enc-Enc	Enc-Dec	Dec-Dec
Transformer	-0.47	-1.05	-0.32
-DropHead	-0.31	-0.79	-0.27
-Scheduled DropHead	-0.28	-0.70	-0.25
Transformer (more heads)	-0.41	-0.92	-0.29
-DropHead	-0.25	-0.61	-0.24
-Scheduled DropHead	**-0.21**	**-0.54**	**-0.20**

Table 4: Average variation of the BLEU score after removing the dominant attention head in different models.

Dominant head analysis The aforementioned experiments on machine translation and text classification datasets demonstrates that our proposed method is able to improve the performance of transformer models. However, as the performance improvement may exclusively come from the better regularization effect yielded by DropHead, it's still unclear whether the DropHead mechanism can prevent the MHA mechanism from being dominated by a few attention heads.

Therefore, we conduct an experiment to analyze the influence of the dominant head in each transformer layer on the final performance of the transformer model. Specifically, we evaluate the model's performance by masking one head at each time and the dominant head is then determined as the one without which the performance of the model degrades the most. We measure the average BLEU score variation of the model's performance after removing the dominant attention head in each layer. Following Michel et al. (2019), we report the results when removing encoder-encoder attention heads, decoder-decoder attention heads, and encoder-decoder attention heads separately as they may behave differently.

The results are presented in Table 4. We find that the performance degradation when removing the dominant head is significantly reduced when trained with Scheduled DropHead. This suggests that our proposed method can substantially reduce the influence of dominant attention heads and exploit multiple attention heads more effectively. This explains the performance improvement yielded by Scheduled DropHead. In addition, we find that training with Scheduled DropHead can reduce the influence of the dominant heads in the transformer model with more attention heads more effectively, which may explain the improved performance with the transformer model with more attention heads trained with DropHead.
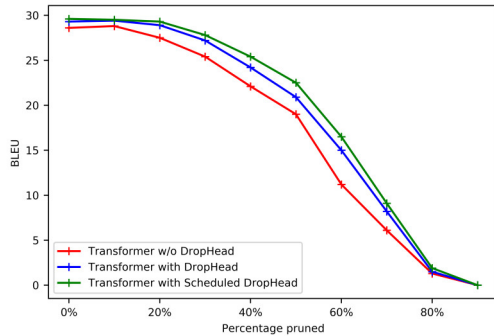
Figure 3: Evolution of BLEU score when attention heads are pruned from different variants of transformer models trained on WMT14 en-de dataset.

Attention head pruning   Similar to other structured dropout methods ([Fan et al., 2019](#); [Ghiasi et al., 2018](#)), training with Scheduled DropHead makes the MHA module more robust to perform the task with missing attention heads. This may make it easier to prune attention heads in the trained transformer model at test time to reduce the number of parameters and improve the model's efficiency during inference. To validate this assumption, we perform attention head pruning with estimated head importance score I_n as described in ([Michel et al., 2019](#)). Specifically, we prune different portion of the total number of attention heads in the transformer-big model trained with and without the DropHead mechanism and the proposed dropout rate schedule, and report the evolution of BLEU score in Table 3. We can see that both the DropHead mechanism and the dropout rate scheduler can increase the model's robustness to attention head pruning. This allows us to prune more attention heads in a transformer model to reduce the parameters in the model and improve the inference efficiency without significantly sacrificing the model's performance.

4.4   Ablation study

We also conduct an ablation study to better understand DropHead and the dropout rate schedule. We analyze the relationship between standard Dropout and Scheduled DropHead; the effect of Scheduled DropHead on different types of attention heads; and the effect of different variants of dropout rate schedules. The experiments are conducted on the machine translation task with WMT14 en-de dataset.

Attention Dropout vs Scheduled DropHead
To analyze the relationship between Scheduled DropHead and the standard Dropout applied to the self-attention layers in the transformer model, we present the performance of transformer models trained by applying standard Dropout, Scheduled DropHead, and the combination of them on the MHA modules in Table 5. We can see that applying standard dropout on the MHA modules only marginally improve the model's performance. In contrast, the proposed Scheduled DropHead can yield significant improvement upon the baseline model and the model trained with standard dropout. We also find that the combination of the standard dropout and DropHead does not yield further improvement. This suggests that Scheduled DropHead is more effective than the standard Dropout for the MHA mechanism.

Model	BLEU
Transformer	28.7
-+ Attention Dropout	28.7
-+ Scheduled DropHead	**29.4**
-+ Combination	**29.3**

Table 5: BLEU score of transformer-big models trained with different regularization methods.

Influence on different types of attention heads
We also analyze the effect of Scheduled DropHead on different types of attention heads. Specifically, we apply Scheduled DropHead separately on encoder-encoder attention, encoder-decoder attention, and decoder-decoder attention modules. The results are presented in Table 6. We can see that the proposed approach is more effective when used on encoder-encoder self-attention and encoder-decoder attention modules, which is consistent with the finding of ([Michel et al., 2019](#)).

Comparison between different dropout rate schedules   We then analyze the influence of different dropout rate schedules by comparing the proposed dropout schedule with two variants. As illustrated in Figure 2, the first variant (curriculum dropout schdule) is that used in Scheduled-DropPath ([Zoph et al., 2018](#)) which linearly increases the dropout rate from 0 to the pre-defined dropout rate. The second (anti-curriculum dropout schdule) is the opposite of the former schedule which linearly decreases the dropout rate from

Model	BLEU
Transformer	
-w/o Scheduled DropHead	28.6
-Enc-enc Scheduled DropHead	28.9
-Enc-dec Scheduled DropHead	29.1
-Dec-dec Scheduled DropHead	28.7
-Scheduled DropHead	**29.4**

Table 6: BLEU score of transformer-big models trained with Scheduled DropHead applied on different MHA modules.

Model	BLEU
Transformer	
-DropHead (constant dropout rate)	29.2
-Curriculum DropHead	29.0
-Anti-curriculum DropHead	28.7
-Scheduled DropHead	**29.4**

Table 7: BLEU score of transformer-big models trained with Scheduled DropHead with different dropout rate schedules.

the pre-defined dropout rate to 0. The result is shown in Table 7. We can see that both curriculum dropout rate scheduler and anti-curriculum dropout schedule fails to improve the performance of the vanilla DropHead where the dropout rate is constant, which demonstrates the effectiveness of the proposed dropout rate schedule.

5   Related work

Since its introduction, dropout (Srivastava et al., 2014) has inspired a number of regularization methods for neural networks, including Drop-Connect (Wan et al., 2013), Maxout (Goodfellow et al., 2013), Spatial Dropout (Tompson et al., 2015), StochasticDepth (Huang et al., 2016), DropPath (Larsson et al., 2016), Drop-Block (Ghiasi et al., 2018), LayerDrop (Fan et al., 2019), etc. While standard dropout works well for fully-connected neural networks, structured dropout (Huang et al., 2016; Larsson et al., 2016; Ghiasi et al., 2018) that randomly drops an entire channel or input block appears to work better in convolutional neural networks. For RNNs, Variational Dropout (Gal and Ghahramani, 2016), ZoneOut (Krueger et al., 2016), and Word Embedding Dropout in both word-level (Gal and Ghahramani, 2016) and embedding-level (Zhou

et al., 2019) are the most widely used methods. More recently, there exists work investigates employing standard Dropout on attention layers (Zehui et al., 2019), or applying StochasticDepth on transformer layers (Fan et al., 2019). However, to the best of our knowledge, regularization approaches specifically designed for the MHA mechasim in the transformer architecture are currently under-explored.

DropHead is also closely related to Spatial Dropout (Tompson et al., 2015), which is a structured dropout method for convolutional neural networks that randomly drops an entire channel from a feature map. The proposed dropout rate scheduler is related to the Curriculum Dropout and the dropout rate scheduler in ScheduledDropPath and DropBlock. Finally, our work is also inspired by recent work analyzing the role of one attention head in the MHA mechanism (Michel et al., 2019; Voita et al., 2019) which found that most attention heads in MHA fail to contribute to the performance of the transformer model and can be effectively pruned at test time.

6   Conclusion

In this work, we introduce Scheduled DropHead, a simple regularization approach for training transformer models. DropHead is a form of structured dropout that drops an entire attention head in the multi-head attention mechanism, which prevents MHA from being dominated by a few heads, reduces excessive co-adaptation between attention heads, and facilitates attention head pruning. We also propose a specific dropout rate scheduler that is motivated by the training dynamic of the MHA mechanism to improve the performance of the vanilla DropHead. Our experiments on machine translation and text classification benchmarks demonstrate that the DropHead mechanism and the dropout rate scheduler can effectively improve the performance of a competitive transformer model. We also find that the proposed approach can reduce the influence of the dominant attention head and improve the model's robustness to attention head pruning.

For future work, we plan to apply DropHead on pretraining language models such as BERT as well as other natural language generation tasks such as text summarization and dialogue systems.

References

Karim Ahmed, Nitish Shirish Keskar, and Richard Socher. 2017. Weighted transformer network for machine translation. *arXiv preprint arXiv:1711.02132*.

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

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

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

Angela Fan, Edouard Grave, and Armand Joulin. 2019. Reducing transformer depth on demand with structured dropout. *arXiv preprint arXiv:1909.11556*.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in neural information processing systems*, pages 1019–1027.

Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. 2018. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10727–10737.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. *arXiv preprint arXiv:1302.4389*.

Tianyu He, Xu Tan, Yingce Xia, Di He, Tao Qin, Zhibo Chen, and Tie-Yan Liu. 2018. Layer-wise coordination between encoder and decoder for neural machine translation. In *Advances in Neural Information Processing Systems*, pages 7944–7954.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. 2016. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer.

Rie Johnson and Tong Zhang. 2017. Deep pyramid convolutional neural networks for text categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 562–570.

David Krueger, Tegan Maharaj, János Kramár, Mohammad Pezeshki, Nicolas Ballas, Nan Rosemary Ke, Anirudh Goyal, Yoshua Bengio, Aaron Courville, and Chris Pal. 2016. Zoneout: Regularizing rnns by randomly preserving hidden activations. *arXiv preprint arXiv:1606.01305*.

Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Fractalnet: Ultra-deep neural networks without residuals. *arXiv preprint arXiv:1605.07648*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are sixteen heads really better than one? *Advances in neural information processing systems (To appear)*.

Pietro Morerio, Jacopo Cavazza, Riccardo Volpi, René Vidal, and Vittorio Murino. 2017. Curriculum dropout. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3544–3552.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to fine-tune bert for text classification? In *China National Conference on Chinese Computational Linguistics*, pages 194–206. Springer.

Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. 2015. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656.

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.

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. *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics.*

Ellen M Voorhees and Dawn M Tice. 1999. The trec-8 question answering track evaluation. In *TREC*, volume 1999, page 82. Citeseer.

Li Wan, Matthew Zeiler, Sixin Zhang, Yann Le Cun, and Rob Fergus. 2013. Regularization of neural networks using dropconnect. In *International conference on machine learning*, pages 1058–1066.

Yingce Xia, Tianyu He, Xu Tan, Fei Tian, Di He, and Tao Qin. 2019. Tied transformers: Neural machine translation with shared encoder and decoder. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 5466–5473.

Lin Zehui, Pengfei Liu, Luyao Huang, Jie Fu, Junkun Chen, Xipeng Qiu, and Xuanjing Huang. 2019. Dropattention: A regularization method for fully-connected self-attention networks. *arXiv preprint arXiv:1907.11065.*

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in neural information processing systems*, pages 649–657.

Wangchunshu Zhou, Tao Ge, Ke Xu, Furu Wei, and Ming Zhou. 2019. Bert-based lexical substitution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3368–3373.

Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. 2018. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710.