

Dear Editors and Reviewers:

Thanks very much for taking the time to review this manuscript. We appreciate all your comments and suggestions. We provide a detailed response to all of the previous reviews, as well as an updated manuscript highlighting in [BLUE](#).

Thanks again.

1. RESPONSE TO REVIEWER #1

Comment 1: From my understanding, this work focuses on hidden semantemes instead hidden dimensions. I suggest that the authors reconsider the title of the paper.

Response: In fact, the reviewer caught the focus of the manuscript and we accept suggestions for rename the title thankfully. The new title is called "Pay Attention to the Hidden Semanteme."

Modification: **Main heading** is renamed as "[Pay Attention to the Hidden Semanteme](#)".

Comment 2: Since X_{dt} carries the token bias, why not use the original token X instead of the sampled token X_w in Eq2? In this way, the bias carried by X_{dt} is richer, and the calculation is lower.

Response: Although the using of original token representation can obtain richer information, it will also lead to the convergence of the mean and variance of row vectors in matrix X_{dt} , which will increase the complexity of classification. Thus, we proposed the Binormal Sampling Method to sparse the representation of original tokens, which is an anti-masking mechanism and different from not only traditional sampling methods but also the masking mechanism in Transformers or BERTs. The feature maps of traditional sampling methods, e.g. up-sampling or down-sampling methods, are on different scales before and after the sampling operation, but those of the Binormal Sampling method are same. The dynamic masking mechanism masks tokens at the same location in each epoch, but the Binormal Sampling masks different tokens in every layer and epoch. In other words, the masking probability of a token in the Dynamic Masking mechanism is p , but it is $(1 - p)^j$ in the j^{th} Binormal Sampling layer. Hence, the usage of the sparse X_w could results in the separation of the mean and variance of row vectors in matrix X_{dt} .

Comment 3: Since matrix X_{dt} represents the relationship among hidden layers, why X_{dt} can be regarded as the direct topic?

Response: We appreciate the vital question of the reviewer that how \mathbf{X}_{dt} worked as the direct topic. The traditional word embedding representation includes the word into the high-dimensional vector space, which is easier to find the category segmentation plane, but the axis in the word vector space has no special meaning. If the axis of the word vector space represents the topic, then the word vector space is the topic space, which means that each token has its own topic embedding representation. Since the usage of abstracting feature of the convolutional network in the n-grams way, the output of the convolutional layer could be regarded as the local topic \mathbf{X}_{at} in Eq2. Furthermore, using sampling result \mathbf{X}_w weighted local topic \mathbf{X}_{at} could let the representation preferred to a specific token. That's why we called the \mathbf{X}_{dt} direct topic. We have added the related reason in Section 3.2, the modification is as follows:

Modification: The traditional word embedding representation includes the word into the high-dimensional vector space, which is easier to find the category segmentation plane, but the axis in the word vector space has no special meaning. If the axis of the word vector space represents the topic, then the word vector space is the topic space, which means that each token has its own topic embedding representation. Based on this idea, the HBA mechanism is proposed as a special feature representation method, to use specific token weighted topic as the direct topic. The HBA are computing in three steps:

1. Obtain a sparse representation \mathbf{X}_w of tokens by the Binomial Sampling layer (see Algorithm 1).
2. Using the convolutional layer to get the ambiguous topic \mathbf{X}_{at} in the n-grams way and weighting it by \mathbf{X}_w . The result \mathbf{X}_{dt} is the direct topic representation with specific token preference.
3. Obtain the global information \mathbf{X}_{ag} by a fully-connected layer the dot \mathbf{X}_{ag} and \mathbf{X}_{dt} as direct global information, which is also the final representation of the HBA.

Comment 4: The number of self-attention parameters is $o(n^2d)$ in Tab.1. However, the total number of self-attention parameters is the sum of all fully-connected layer parameters, so the number of self-attention parameters should be $o(nd^2)$. Thus, I would appreciate the authors check the complexity in Tab.1 for correctness.

Response: Thank you for your attention to the details of the paper. We have corrected the relevant typo and update the detail of the manuscript.

Modification:

Table 1

The comparison of the FLOPs and the parameters scale among the fully-connected layer, the token-mixing layer, the self-attention layer, the convolutional layer, the grouped convolutional layer, the fast-fourier transform and our HBA and multi-head HBA layer. n is the sequence length. d means the number of the hidden dimension. k is the kernel width of convolution. g means the number of groups of the kernels. h represents the number of the heads. Here, $k < h < g < d < n$.

Layer Type	Parameters	FLOPs
fully-connected	$\mathcal{O}(d^2)$	$\mathcal{O}(n \cdot d^2)$
token-mixing	$\mathcal{O}(n^2)$	$\mathcal{O}(n^2 \cdot d)$
self-attention	$\mathcal{O}(d^2)$	$\mathcal{O}(n^2 \cdot d)$
1D-convolution	$\mathcal{O}(k \cdot n^2)$	$\mathcal{O}(k \cdot n \cdot d)$
grouped convolution	$\mathcal{O}(\frac{k \cdot n^2}{g})$	$\mathcal{O}(\frac{k \cdot n \cdot d}{g})$
fast-fourier transform	$\mathcal{O}(1)$	$\mathcal{O}(n \cdot d \cdot \log(n \cdot d))$
HBA	$\mathcal{O}(d^2)$	$\mathcal{O}(n \cdot d^2)$
multi-head HBA	$\mathcal{O}(\frac{d^2}{h})$	$\mathcal{O}(\frac{n \cdot d^2}{h})$

Comment 5: The experiments are quite adequate, but why are the results of the MHBA-Mixer-64d model on amazon in Tab.8 and hyperpartisan in Tab.9 not presented?

Response: Due to the device limitation, the training time of the MHBA-Mixer-64d on amazon is very long, we haven't obtained the result by the time we submit the manuscript. Now we added the result in tables. As for the result on hyperpartisan in Tab.9, the feature map can not be divisible by groups. We merged Tab.8, Tab.9 and Tab.10 to a new table and update the description of the table, the modification is as follows:

Modification:

Table 8

The accuracy and number of parameters of different models on each public data sets. To reduce the size of the layout, some dataset names are abbreviated, such as "AGN." means AGNews, "Ama." means Amazon, "DBp." means DBpedia and "Hyp." means Hyperpartisan. Top-3 scores are colored by red, gold and green.

Model	Accuracy (%)									Param. (M)
	AGN.	Ama.	DBp.	Hyper.	IMDb	Yelp-2	SST.	CoLA	QQP	
RoBERTa	/	/	/	87.40	95.30	/	96.70	67.80	90.20	125
XLNet	95.55	/	99.40	/	96.21	98.63	94.40	69.00	90.40	240
Bert Large	/	97.37	99.36	/	95.49	/	93.70	/	/	340
UDA	/	96.50	98.91	/	95.80	97.95	/	/	/	340
Bert-ITPT-FiT	95.20	/	99.32	/	/	/	/	/	/	340
gMLP	/	/	/	/	/	/	94.80	/	/	365
Longformer	/	/	/	94.80	95.70	/	/	/	/	149
FNet	/	/	/	/	/	/	94.00	67.00	85.00	85
MobileBERT	/	/	/	/	/	/	92.80	51.10	70.50	25.3
pNLP-Mixer	91.03	93.50	98.40	89.20	82.90	91.70	80.90	69.94	84.90	5.3
HyperMixer	/	/	/	/	/	/	80.70	/	83.70	12.5
MHBA-Mixer-64d-1h-2l	91.38	91.28	93.49	77.86	86.79	92.81	80.21	69.12	81.55	0.13
MHBA-Mixer-64d-16h-2l	91.68	91.17	98.11	/	87.08	92.35	83.21	69.23	81.96	0.10
MHBA-Mixer-256d-16h-2l	91.79	91.88	98.44	89.43	87.88	92.57	83.48	69.51	82.02	0.73

In this section, the proposed MHBA-Mixers will complete with both SOTA models and counterparts mention in Table 3 on 9 public data sets referred in Table 2. Results are as Table 8.

In terms of the number of parameters, the proposed MHBA-Mixers have the lowest number of parameters, the pNLP-Mixer ranks 2 and the HyperMixer ranks 3. Compared with large pre-trained models, the quantity of parameters of the MHBA-Mixer-64d-16h-2l is very less, i.e., $1250\times$ fewer than its of RoBERTa, $2400\times$ fewer than its of XLNet and $3400\times$ fewer than its of BERTs. As for counterparts, the number of parameters of the pNLP-Mixer and the HyperMixer is also greater than its of MHBA-Mixer-256d-16h-2l which is the largest model of MHBA-Mixers.

In model quality, MHBA-Mixers also have highlights though there is a gap with large pre-trained models. On AGNews, the MHBA-Mixer-256d-16h-2l achieves a 91.79% accuracy, ranks 3, only 3.76% lower than XLNet and 3.41% lower than Bert-ITPT-FiT. But on IMDb, the MHBA-Mixer-256d-16h-2l achieves 87.88% on accuracy, 4.92% higher than its of the pNLP-Mixer. Most commendably, our MHBA-Mixer-256d-16h-2l even outperforms RoBERTa on Hyperpartisan, and also outperform the pNLP-Mixer on Yelp-2 and SST-2 by a great gap. On CoLA, our MHBA-Mixers show their effectiveness by comparing with other pre-trained models. As for QQP, the accuracy of the MHBA-Mixer-256d-16h-2l is 11.52% higher than its of the MobileBERT. Comparing with the HyperMixer, our MHBA-Mixer is better than it on SST-2 but little worse on QQP. From the perspective of the model accuracy on 9 datasets, MHBA-Mixers outperform the pNLP-Mixer on 6 tasks, better than the MobileBERT on 2 tasks (only 3 tasks) and the HyperMixer on 1 task (only 2 tasks). Thus, MHBA-Mixers work well than counterparts.

Overall, while there is still a gap with SOTA models, MHBA-Mixers have huge advantages in model size. Compared to counterparts, MHBA-Mixers are better than them in both model size and model quality. Moreover, on some datasets, the gap between MHBA-Mixers and pre-trained models is not very large.

Comment 5: This paper would also benefit from proofreading for a few typos and grammar mistakes.

Response: Thanks for advices. We proofread our manuscript for the spelling, coefficients, functional, notations and grammar issues in this revision, and update the figures accordingly.

2. RESPONSE TO REVIEWER #2

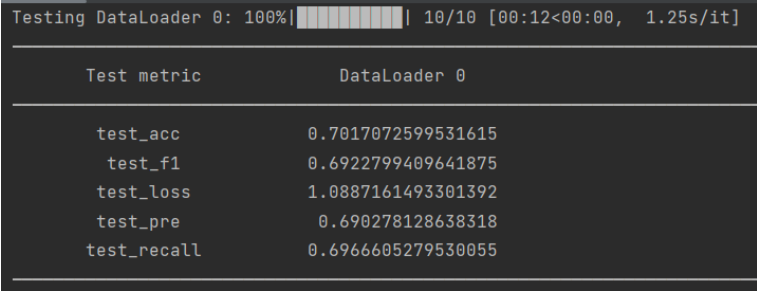
Comment 1: The paper, however, does not link well with recent literature on sentiment analysis appeared in relevant top-tier journals, e.g., the IEEE Intelligent Systems department on "Affective Computing and Sentiment Analysis". Also, new trends on neurosymbolic AI for explainable sentiment analysis are missing.

Response: Sentiment analysis is a field with high research value, and the reviewer's suggestions are also very pertinent, so we decided to refer to the recommended literature in the introduction. In this paper, the model we propose is a general lightweight model, which should be able to handle multiple downstream tasks. Of course, sentiment analysis is also included in these tasks. This suggestion gives us new inspirations, and we will further optimize the proposed model and further explore its application in affective computing.

Comment 2: Authors seem to handle sentiment analysis mostly as a binary classification problem (positive versus negative). What about the issue of neutrality or ambivalence? Check relevant literature on detecting and filtering neutrality in sentiment analysis and recent works on sentiment sensing with ambivalence handling. Finally, the manuscript only cites a few papers from 2022: check latest works on aspect-based sentiment analysis via graph convolutional networks, meta-based self-training for sentiment analysis, and prompt-based sentiment analysis and emotion detection.

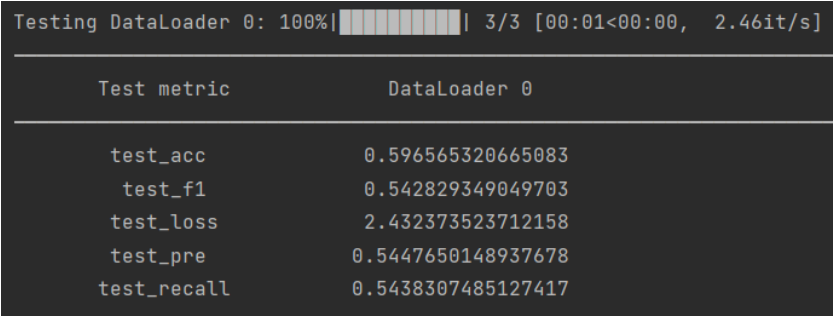
Response: As the response of Comment 1, the sentiment analysis is one of downstream tasks our model can process. Our team members have set up the finetune our general lightweight model to process ABAS. The comparison among graph convolutional networks methods, meta-based self-training methods and prompt-based methods will be highlighted. To demonstrate feasibility, we added more experiments in sentiment analysis data sets including Semeval_2014_task_1 (3 classes "NEUTRAL", "ENTAILMENT" and "CONTRADICTION") and Tweet_Eval_Emotion (4 classes "angry", "joy", "optimism", "sadness"). Results are as follows:

Fig1.The Result on Semeval_2014_task_1



Testing DataLoader 0: 100% ██████████ 10/10 [00:12<00:00, 1.25s/it]	
Test metric	DataLoader 0
test_acc	0.7017072599531615
test_f1	0.6922799409641875
test_loss	1.0887161493301392
test_pre	0.690278128638318
test_recall	0.6966605279530055

Fig2. The result on Tweet_Eval_emotion



Testing DataLoader 0: 100% ██████████ 3/3 [00:01<00:00, 2.46it/s]	
Test metric	DataLoader 0
test_acc	0.596565320665083
test_f1	0.542829349049703
test_loss	2.432373523712158
test_pre	0.5447650148937678
test_recall	0.5438307485127417

As shown in Fig1 and Fig2, the results on these two datasets are not good, but we will continue to improve for the focus area of Sentiment Analysis including Aspect-based Sentiment Analysis in future work.

Comment 3: The manuscript presents some bad English constructions, grammar mistakes, and misuse of articles: a professional language editing service (e.g., the ones offered by IEEE, Elsevier, or Springer) is strongly recommended in order to sufficiently improve the paper's presentation quality for meeting the high standards of INS. Finally, double-check both definition and usage of acronyms: every acronym, e.g., NLP, should be defined only once (at the first occurrence) and always used afterwards (except for abstract and section titles). Also, it is not recommendable to generate acronyms for multiword expressions that are shorter than 3 words, e.g., CV (unless they are universally recognized, e.g., AI).

Response: We agree and appreciate the reviewer's suggestions on language expression. We plan to find a team of professional editors (as mentioned above) in the future to further improve our manuscript. In the current revision, we have asked native English writers for help in correcting grammatical and syntactic errors.

3. RESPONSE TO REVIEWER #3

Comment 1: The paper does good work explaining the related work but needs to add more background discussion on heavy attention and mixer models

Response: Thanks for the comment, we reorganized the **Related Work** and added more discussion on heavy attention with pruning, low-rank factorization and quantization. The modification is as follows:

Modification:

Here, the discussion on lightweight models is organized into two parts: (1) Optimizations of self-attention mechanism. (2) Alternative by lightweight Mixer modules.

Optimizations In this line, technologies of pruning, low-rank factorization and Quantization are used to reduce complexity popularly. For example, in **Chen's** work, they pruned weight matrix of self-attention layers and outperformed than Pytorch, TensorRT, and FasterTransformer. However, the final model needs to be adjusted several times, which is a complicated process and higher time require. **Wang et al.** proposed the SPAtten to cascade head pruning, cascade head pruning and progressive quantization, but appropriate thresholds are difficult to obtain and require extensive experience. Q8Bert compressed the BERT model by 4x with minimal accuracy loss, but also faced the challenge on deploying on different machines. In this context, Mixer modules are designed to substitute for the self-attention layer.

Mixer Modules Targeting to different objects, the mixers can be grouped into token-mixing and spatial-mixing.

Comment 2: The paper proposes hidden bias attention as an alternative to self-attention model to reduce the parameters and make it faster. The paper should also discuss different optimizations proposed to speed up the self-attention layer ([1], [2]) and compare/discuss their performance with the proposed hidden bias attention.

[1.] Shiyang et al., E.T.: Re-Thinking Self-Attention for Transformer Models on GPUs, SC'21

[2.] Hanrui Wang et al., SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning, HPCA'21

Response: We have added the discussion of reviewer recommended methods in Related Work. In our experiment, the experimental results of the large pre-training

model are used to describe the upper bound of the index of the data set. Comparison with counterparts, such as FNet, pNLP-Mixer and HyperMixer, is the basis for evaluating the effectiveness of our method. Methods recommended by the reviewer are modifications based on the large pre-training model. Our method is to redesign the model without using the pre-training technology. Due to an order of magnitude gap in the amount of training data, the comparison is unfair we think. However, we pay full attention to the reviewer's suggestions and will fairly compare the differences between the above methods and ours in subsequent studies.

Comment 3: The paper states that using higher depth for hidden dimension is not much helpful for processing short text but has some usage scenarios. But the usage scenarios are not clearly stated.

Response: According to Tab.7, the accuracy of model does increase with hidden dimension, but not very significant. That's why we said greater hidden dimensions are not helpful to increase the income of the MHBA-Mixer. Compared with the MHBA-Mixer-256d, the MHBA-Mixer-64 has fewer parameters and their accuracy is about the same. Therefore, **we recommend using the lower dimensional MHBA-Mixer-64d instead of MHBA-Mixer-256d, unless higher accuracy is required.**

Modification: Table 7 shows that it's not obvious that the income increases with the hidden dimension. In other words, the accuracy does increase faintly with hidden dimensions, but the parameters increased more substantially. **Therefore, we recommend using the lower dimensional MHBA instead of higher one for processing short text, unless higher accuracy is required.**

Comment 4: A lot of optimizations like pruning and quantization are proposed to speed up self-attention layer with negligible accuracy loss. Do HBAs have the same robustness against pruning and quantization?

Response: In theory, such robustness exists. Since we have discussed in the manuscript that the influence of the depth of MHBA-Mixer model on the model performance is not obvious, the structured pruning does not improve the performance of models using HBA/MHBA. Secondly, we believe that the quantization model is feasible, which only changes the data representation precision of the model. However, whether the accuracy of the quantified model will be improved needs to be verified in subsequent studies.

Comment 5: The paper introduces two MHBA-Mixer models (64d and 256d) in Section 4.3 but the difference is not discussed.

Response: We added naming rules in Section 4.1 of the manuscript. The MHBA-Mixer-64d and the MHBA-Mixer-256d are different in number of hidden dimensions using. Details are as follows:

Modification: Here, we named of MHBA-Mixer with MHBA-Mixer-d-h-l, where d means the number of hidden dimensions, h is the number of heads, l denotes the number of mixer layers.

Comment 6: In Section 4.2, how is self-attention modified as a variant of HBA? More discussion would be helpful.

Response: We add the description of how is self-attention modified as a variant of HBA. Additionally, the modification is reversible.

Modification:

Table 5

The comparisons of different structures of the HBA in Accuracy (%).

Datasets	Attentions			
	self-attention	self-attention q_{bs}	self-attention k_{gc}	HBA
AGNews	91.44	91.50	91.63	91.30
IMDb	86.23	85.53	86.50	87.80
SST-2	80.73	81.10	81.97	82.10

¹ Self-attention q_{bs} and self-attention k_{gc} are the variations of the Self-attention and HBA. Notice, when the HBA take place both binomial sampling layer and grouped convolutional layer to fully-connected layers, the HBA is same to the self-attention layer.

Comment 7: Does the evaluation show that the performance of the mixer model can be good even with less number of parameters? How about the training data? Do they require more/less training data than the large models?

Response: We can answer the first question with absolute certainty: Yes! The pre-trained technology hasn't used in this work, so no additional data, other than the training data itself presented in this manuscript. Details of training data are as follows:

Modification:

Table 2

This paper uses following public data sets. () represents the max sequence length.

Tasks	Datasets(Max Sequence Length)		
Text Categorization	AGNews(128)	Amazon-2(128)	DBpedia(128)
Semantic Analysis	Hyperpartisan(2048)	IMDb(1024)	Yelp-2(128)
Natural Language Inference	SST-2(128)	CoLA(128)	QQP(128)
Attention Comparison*	AGNews(128)	IMDb(128)	SST-2(128)

¹ Attention Comparison* is the exploring task of which structure of the HBA is best.

This paper uses the most popular tasks including AGNews¹ (120K training examples, 4 classes), Amazon-2² (36M training examples, 2 classes), DBpedia³ (560K training examples, 14 classes), IMDb⁴ (2.5K training examples, 2 classes), Yelp-2⁵ (560K training examples, 2 classes), Hyperpartisan⁶ (645 examples, 2 classes), QQP, SST-2, and CoLA on the GLUE⁷ benchmark set. Especially, the IMDb and Hyperpartisan are long text classification tasks, in which samples are longer than 512. The text length will be set to 1024 and 2048, same as pNLP-Mixer, fairly and respectively. The model will train for 60 epochs and use the same data split as pNLP-Mixer. The rest will be set to 256. More details could be found in Table 2.

Comment 8: Multi-head attention has good scalability and is kind of computation friendly. Does HBA-based model have better/similar computation efficiency and scalability? A comparison of inference time would be much better.

Response: Thanks for the advice. We added the comparison among the Multi-Head Attention layer, Fnet layer, Mixer layer and Multi-Head HBA on parameters and FLOPs. Details are as follows:

Modification:

4.3 The comparison with other methods

¹ https://huggingface.co/datasets/ag_news

² https://huggingface.co/datasets/amazon_polarity

³ https://huggingface.co/datasets/dbpedia_14

⁴ <https://huggingface.co/datasets/imdb>

⁵ https://huggingface.co/datasets/yelp_polarity

⁶ https://huggingface.co/datasets/hyperpartisan_news_detection

⁷ <https://huggingface.co/datasets/glue>

In this section, the Mult-Head HBA (MHBA) will be compared with the Multi-Head Attention (MHA) layer, Fnet Layer, Mixer Layer on Parameters (Param.) and FLOPs. Results are as follows:

Table 9

The comparison among the Multi-Head Attention layer (MHA), Fnet layer, Mixer layer and Multi-Head HBA.

Layer Type	Param.(M)	FLOPs(G)
MHA (Devlin et al., 2019)	12.596	6.447
FNet (Lee-Thorp et al., 2022)	8.398	4.299
Mixer (Fusco et al., 2022)	0.559e-3	6.763e-2
Multi-Head HBA(Ours)	0.010e-3	3.079e-3

As shown in Table 9, our Multi-Head HBA has the lowest parameters and FLOPs, which can reflect the effectiveness of our method directly.

Thanks for all the valuable reviews which help us improve the quality of our manuscript and give us more insights into the research.

Best regards,

Huanling Tang, Xiaoyan Liu, Yulin Wang, Quansheng Dou, Mingyu Lu

2023.2.10