On the Role of Hidden States of Modern Hopfield Network in Transformer

Tsubasa Masumura* **Masato Taki***
Graduate School of Artificial Intelligence and Science, Rikkyo University, Japan
{25wr001m,taki_m}@rikkyo.ac.jp

Abstract

Associative memory models based on Hopfield networks and self-attention based on key-value mechanisms have been popular approaches in the study of memory mechanisms in deep learning. It has been pointed out that the state update rule of the modern Hopfield network (MHN) in the adiabatic approximation is in agreement with the self-attention layer of Transformer. In this paper, we go beyond this approximation and investigate the relationship between MHN and self-attention. Our results show that the correspondence between Hopfield networks and Transformers can be established in a more generalized form by adding a new variable, the hidden state derived from the MHN, to self-attention. This new attention mechanism, modern Hopfield attention (MHA), allows the inheritance of attention scores from the input layer of the Transformer to the output layer, which greatly improves the nature of attention weights. In particular, we show both theoretically and empirically that MHA hidden states significantly improve serious problem of deep Transformers known as rank collapse and token uniformity. We also confirm that MHA can systematically improve accuracy without adding training parameters to the Vision Transformer or GPT. Our results provide a new case in which Hopfield networks can be a useful perspective for improving the Transformer architecture.

1 Introduction

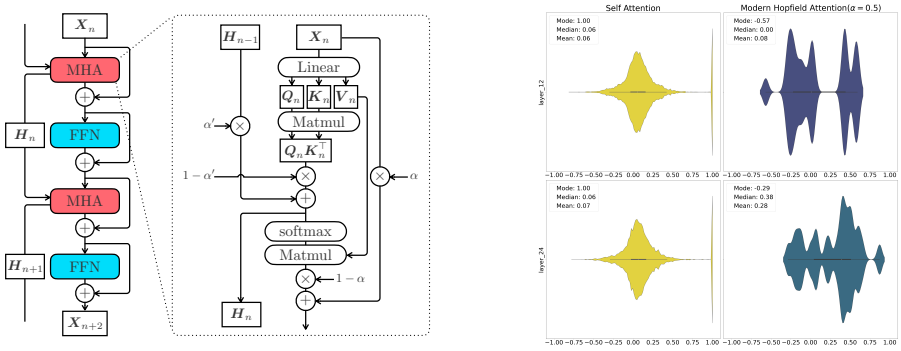
The relationship between associative memory in Hopfield networks [22, 1], which has attracted interest from neuroscientists, and Transformers [46] based on key-value memory that have been studied in machine learning has attracted interest from the research community [45, 42, 3, 48, 21]. One of the most interesting results is the finding in [40, 31] that translating modern Hopfield networks into neural networks yields the Transformer architecture that has been very successful in natural language processing [38, 9] and computer vision [12]. What, then, do more general modern Hopfield networks imply for deep learning? This paper gives a concrete answer to this question.

Hopfield networks [22, 1] are a class of models for associative memory. Despite these interesting properties, classical Hopfield networks have the limitation of small storage capacity. Recently, [30] proposed Dense Associative Memory which can achieve storage capacity that scales exponentially or power-wise with respect to the number of neurons by introducing high nonlinearity [7]. These models with large storage capacity are collectively referred to as modern Hopfield networks [40, 31].

Recent advances in Transformer architecture, including its application to language models, have led to significant advances in the study of self-attention mechanisms. These advances have also shed new light on Hopfield networks. In [40], it was shown that the state update rules of modern continuous Hopfield networks (MCHNs) have a mathematical structure exactly equivalent to that of the self-attention mechanism. Furthermore, [31] developed this relationship theoretically, pointing out that

*These authors contributed equally to this work

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) (b)

Figure 1: (a) The left figure shows the layer structure of Transformer architecture using modern Hopfield attention (MHA). As the hidden state $\boldsymbol{H}_n$ propagates through each attention layer, information from the upper layer's attention scores is reused in the lower layers. Attention score $\boldsymbol{Q}_n \boldsymbol{K}_n^\top$ is accumulated in the hidden state of each layer, and this value is used for attention calculation. (b) A visualization of the token uniformity in layers 12 and 24 of GPT-2 (Medium) trained on the Wikitext103 dataset, showing a violin plot of the cosine similarity between the tokens. For GPT-2 in the left column, there is a strong peak at similarity 1, and both layers have a mode of 1. On the other hand, in the case of GPT-2 with MHA in the right column, the cosine similarity is kept low and the uniformity of the tokens is dramatically improved.

the self-attention layer of Transformer coincides with the adiabatic limit of generic modern Hopfield networks. Thus, it is expected that there is a deep relationship between the Transformer's architectural design and associative memory. Still the connection, however, lack fundamental understanding.

Therefore, we consider the question: is it possible to interpret modern Hopfield networks without adiabatic approximation in terms of Transformers? The adiabatic limit approximation removes the hidden state dynamics from MCHN. In this paper, we show that maintaining this dynamics introduces a hidden state on the Transformer side, thereby creating a mechanism for the propagation of attention score information from the upper to the lower layers.

By implementing this new attention mechanism into Transformer architectures, this paper introduces a new type of self-attention layer called modern Hopfield attention (**MHA**), as shown in Figure 1(a). MHA does not require additional parameters, and the increase in computational complexity is very small. Nevertheless, simply using MHA instead of the usual attention layer, performance gains can be obtained in various natural language processing and image recognition tasks. Furthermore, we found that MHA effectively solves the problem known as rank collapse, or token uniformity, where Transformer's tokens lose diversity as Figure 1(b). These results indicate that ideas derived from the Hopfield network may provide a new perspective for Transformer research.

In summary, our contributions are as follows:

- By investigating the relationship between MCHN and Transformer beyond the adiabatic approximation, we showed that this correspondence can be further generalized. Based on the correspondence with MCHN, we proposed a new type of attention mechanism with hidden state, MHA. The MHA-based Transformer improves the nature of attention weights by sharing attention score information across layers.
- By training Transformers with MHA, we experimentally showed that the MHA mechanism also contributes to the performance of the Transformers. In particular, we investigated image recognition with Vision Transformer and text generation tasks with GPT-2, and confirmed that MHA actually improves their performance. This method does not generate any additional parameters, and thus can lead to performance gains with only a small increase in computational complexity.
- Theoretically and experimentally, we showed that the reason why MHA works so well as an alternative to self-attention is related to rank collapse. The hidden state of MHA is likely to enhance its performance by cleverly improving Transformer's rank collapse as Figure

1(b). Our results suggest that the Hopfield Networks can provide guidance for improving the Transformer architectures.

2 Related Works

The relationship between the modern Hopfield network and the Transformer was investigated in [40, 31], and various improvements and extensions have been made to the modern Hopfield network [34, 53, 26, 25, 41, 5, 21, 24, 23, 49, 19, 17].

In [40], the authors demonstrate the fast convergence of the Modern Hopfield Network (MHN) and justify its use as a conventional module in Transformer-related architectures. On the other hand, in this study's MHA, we propose a dynamic structure that maintains and updates hidden states across layers. More specifically, our MHA naturally incorporates Hopfield recursion into the Transformer layer structure, as "state accumulation and updating" are performed in each layer.

Research on improving the design of Transformer architecture using this relationship has also been conducted in [20]. Unlike these studies, this paper focuses on the effect of keeping hidden state dynamics of MCHN.

In this paper, we saw that hidden states lead to reuse of attention scores across layers. Attention score reuse has been studied [16, 10] from technical perspective, including improvements to the Pre-LN Transformer. These studies are focus only on encoder architecture and do not consider the special combination of moving average with α' and skip connection modification with α as in MHA. On the other hand, this paper showed that the more extended attention mechanisms in MHA can be understood in terms of modern Hopfield networks, and examines its effects including the decoder Transformer. Furthermore, the essential role of MHA is clarified theoretically and experimentally in terms of rank collapse.

3 Method: Transformers from Hopfield Network

In this chapter, we review the methods [40, 31] used to derive Transformer from MCHN and give a careful treatment of discretization, which has been ignored in previous studies. As a result, we show that hidden state of MCHN leads to significant changes in the mechanisms of self-attention.

3.1 Self-Attention Mechanism

Let T be the number of input tokens with dimension d. $\boldsymbol{X}_n \in \mathbb{R}^{T\times d}$ is the feature obtained by concatenating the input token vectors $\boldsymbol{x}_n$ of the n-th attention layer. The attention weight of Transformer is given by the row-wise softmax value of the attention score, which is given by the inner product of the query and the key, and the dot-product self-attention is calculated by weighting and adding the value vectors together as $\boldsymbol{X}_{n+1} = \text{softmax}\left(\boldsymbol{Q}_n \boldsymbol{K}_n^\top\right) \boldsymbol{V}_n$, where the query, key, value are given by linear projections of the input $\boldsymbol{X}_n$ as $\boldsymbol{Q}_n = \boldsymbol{X}_n \boldsymbol{W}_Q$, $\boldsymbol{K}_n = \boldsymbol{X}_n \boldsymbol{W}_K$ and $\boldsymbol{V}_n = \boldsymbol{X}_n \boldsymbol{W}_V$ [46]. Each token vector is a slice $\boldsymbol{x}_n = (\boldsymbol{X}_n)_{t,:}$ of the feature tensor. Then the formula for attention mechanism for each token is $\boldsymbol{x}_{n+1} = \text{softmax}\left(\boldsymbol{q}_n \boldsymbol{K}_n^\top\right) \boldsymbol{V}_n$.

3.2 Modern Continuous Hopfield Network and its Discretization

MCHN is a network model with bipartite graph connectivity connecting two dynamic variables $\boldsymbol{x}$ and $\boldsymbol{h}$. The connections are given by the network's weights $\boldsymbol{W}$, in which the memories to be associated are stored. $\boldsymbol{x}$ is called the visible state or feature neuron, and $\boldsymbol{h}$ is called the hidden state or memory neuron. In the context of the associative memory model, given a collapsed $\boldsymbol{x}$ as an initial configuration, the complete $\boldsymbol{x}$ is reproduced by association through the time evolution of the state. The time evolution of MCHN is given by the following update rule [31]:

$$\tau_v \frac{d\boldsymbol{x}}{dt} = \boldsymbol{f}\left(\boldsymbol{h}\right) \boldsymbol{W}_1^\top - \boldsymbol{x}, \quad \tau_h \frac{d\boldsymbol{h}}{dt} = \boldsymbol{g}\left(\boldsymbol{x}\right) \boldsymbol{W}_2 - \boldsymbol{h}, \tag{1}$$

where $\tau_{v,h}$ are the time constants of the dynamic system[1]. The activation functions $\boldsymbol{f}(\cdot)$ and $\boldsymbol{g}(\cdot)$ are given by the Lagrangian functions $L_{h,v}$ for $\boldsymbol{h}$ and $\boldsymbol{x}$

$$\boldsymbol{f}\left(\boldsymbol{h}\right) = \frac{\partial L_h}{\partial \boldsymbol{h}}, \quad \boldsymbol{g}\left(\boldsymbol{x}\right) = \frac{\partial L_v}{\partial \boldsymbol{x}}. \tag{2}$$

In this paper, the vectors $\boldsymbol{h} = \left(h_a\right)$, $\boldsymbol{f}\left(\boldsymbol{h}\right) = \left(f_a\left(\boldsymbol{h}\right)\right)$, $\boldsymbol{x} = \left(x_i\right)$ and $\boldsymbol{g}\left(\boldsymbol{x}\right) = \left(g_i\left(\boldsymbol{x}\right)\right)$ are all row vectors. In order to see the correspondence with Transformer below, let us derive discrete time counterpart of MCHN. We then discretize this update rule with a finite difference $\Delta t = t_{n+1} - t_n$ as follows

$$\frac{\tau_v}{\Delta t}\left(\boldsymbol{x}_{n+1} - \boldsymbol{x}_n\right) = \boldsymbol{f}\left(\boldsymbol{h}_n\right)\boldsymbol{W}_1^\top - \boldsymbol{x}_n, \quad \frac{\tau_h}{\Delta t}\left(\boldsymbol{h}_{n+1} - \boldsymbol{h}_n\right) = \boldsymbol{g}\left(\boldsymbol{x}_n\right)\boldsymbol{W}_2 - \boldsymbol{h}_n, \tag{3}$$

where $\boldsymbol{x}_n = \boldsymbol{x}(t_n)$ and $\boldsymbol{h}_n = \boldsymbol{h}(t_n)$. Introducing the ratio between the discretization step width and the time constant as $\frac{\Delta t}{\tau_v} = 1 - \alpha$ and $\frac{\Delta t}{\tau_h} = 1 - \alpha'$, we obtain

$$\boldsymbol{x}_{n+1} = \alpha \boldsymbol{x}_n + (1-\alpha)\boldsymbol{f}\left(\boldsymbol{h}_n\right)\boldsymbol{W}_1^\top, \quad \boldsymbol{h}_{n+1} = \alpha' \boldsymbol{h}_n + (1-\alpha')\boldsymbol{g}\left(\boldsymbol{x}_n\right)\boldsymbol{W}_2. \tag{4}$$

In past studies, the effect of the discretization step α was ignored as negligible, but the precise derivation here leads to an interesting modification of Transformer. In this paper, we give empirical and theoretical results in which α and α''s effect is extremely important.

3.3 Adiabatic Limit and Self-Attention

Specific MCHN model is determined by explicitly selecting the Lagrangians. Model B of [31] is given by the following choice of Lagrangians

$$L_h = \log\left(\sum_a e^{h_a}\right), \quad L_v = \frac{1}{2}\|\boldsymbol{x}\|_2^2. \tag{5}$$

These Lagrangians give the activation functions

$$f_a = \mathrm{softmax}\left(h_a\right), \quad g_i = x_i. \tag{6}$$

The adiabatic limit in [31] $\tau_h \approx 0$ implies $\boldsymbol{h}_n = \boldsymbol{x}_n \boldsymbol{W}_2$. The update rule for (6) is then given by

$$\boldsymbol{x}_{n+1} = \alpha \boldsymbol{x}_n + (1-\alpha)\mathrm{softmax}\left(\boldsymbol{x}_n \boldsymbol{W}_2\right)\boldsymbol{W}_1^\top. \tag{7}$$

Translating (7) by the rule $\boldsymbol{q}_n = \boldsymbol{x}_n \boldsymbol{W}_Q$, $\boldsymbol{W}_1^\top = \boldsymbol{X}_n \boldsymbol{W}_V = \boldsymbol{V}$, $\boldsymbol{W}_2^\top = \boldsymbol{X}_n \boldsymbol{W}_K \boldsymbol{W}_Q^\top = \boldsymbol{K}\boldsymbol{W}_Q^\top$ according to [40, 31], we obtain $\boldsymbol{x}_{n+1} = \alpha \boldsymbol{x}_n + (1-\alpha)\mathrm{softmax}\left(\boldsymbol{q}_n \boldsymbol{K}^\top\right)\boldsymbol{V}$, where $\boldsymbol{X}_n$ is the concatenated tensor of the embedding vectors of all tokens $\boldsymbol{x}_{n1}, \cdots, \boldsymbol{x}_{nT}$. When $\alpha = 0$, i.e., $\Delta t = \tau_v$, this update rule is exactly a usual self-attention mechanism in [46]. In the following, we consider general α and α' to investigate the effect of the hidden state dynamics, which are ignored in the adiabatic limit above.

3.4 Hidden State Dynamics and Modern Attention Attention

If the adiabatic limit is not taken and a finite $\frac{\Delta t}{\tau_h}$ is kept, the dynamics of the hidden state is

$$\frac{\tau_h}{\Delta t}\left(\boldsymbol{h}_{n+1} - \boldsymbol{h}_n\right) = \boldsymbol{g}\left(\boldsymbol{x}_{n+1}\right)\boldsymbol{W}_2 - \boldsymbol{h}_{n+1}. \tag{8}$$

In the following, we use the new parameterization $\frac{\Delta t}{\tau_h} = \frac{1-\alpha'}{\alpha'}$ to obtain a simple formula. The dynamics of Model B is then

$$\boldsymbol{x}_{n+1} = \alpha \boldsymbol{x}_n + (1-\alpha)\mathrm{softmax}\left(\boldsymbol{h}_n\right)\boldsymbol{W}_1^\top, \quad \boldsymbol{h}_{n+1} = \alpha' \boldsymbol{h}_n + (1-\alpha')\boldsymbol{x}_{n+1}\boldsymbol{W}_2. \tag{9}$$

Using the same translation rules as before, we get the following novel modification of attention layer

$$\boldsymbol{x}_{n+1} = \alpha \boldsymbol{x}_n + (1-\alpha)\mathrm{softmax}\left(\boldsymbol{h}_n\right)\boldsymbol{V}_n, \quad \boldsymbol{h}_n = \alpha' \boldsymbol{h}_{n-1} + (1-\alpha')\boldsymbol{q}_n \boldsymbol{K}_n^\top. \tag{10}$$

[1] In the following discussion, we do not assume the tying of $\boldsymbol{W}_1$ and $\boldsymbol{W}_2$. This breaking of the symmetry of the memory matrix violates the assumption of monotonically decreasing energy function in the mathematical discussion of [31]. Interpreting the energy function of MHA in asymmetric settings is a very interesting theoretical challenge for future research.

Thus, if the dynamics of the hidden state in the MHN is maintained and mapped to the self-attention layer, a new variable $\boldsymbol{h}$, determined by the value of the attention scores, is added to the self-attention layer. This variable continues to accumulate the value of the attention score in each layer in the form of an exponential moving average across layers. Through this variable, the attention weights of each layer of the Transformer will have a coordinated behavior. In the following, we will investigate the effect of adding this hidden state on the attention layer from the Transformer's perspective. In this paper, this extended attention mechanism with hidden states will be referred to as **M**odern **H**opfield **A**ttention (**MHA**). Compared to the cost $O(dT^2)$ of computing the dot product of self-attention, the computational complexity added by updating the hidden state is about $O(T^2)$. For Transformer that uses more than several hundred dimensions of d, this is a small increase in computational complexity.

4 Empirical Results

In this chapter, we experimentally investigate how the performance of the model changes when MHA is actually used in place of Transformer's self-attention module. We take the Vision Transformer (ViT) as a representative example of an encoder Transformer model and the GPT-2[39] architecture as a representative example of a decoder Transformer model, and confirm that MHA does indeed lead to systematic performance improvements in several experiments.

4.1 Architecture with MHA

In the following, we will focus on the simplest case $\alpha = \alpha'$. It is straightforward to choose both parameter independently, but consider only this case to reduce the hyperparameters. By using our update rule (10) instead of the attention layer, a new tensor called the hidden state $\boldsymbol{H}_\ell$ propagates across the layers. This tensor accumulates the attention score $\boldsymbol{Q}_\ell \boldsymbol{K}_\ell^\top$ in each layer in the form of an exponential moving average. It is not the original attention score that gives the attention weight, but the softmax of the hidden state. At the same time, a skip connection with the weight $(1-\alpha)$ linked to the coefficient α of the exponential moving average of the hidden state is added according to equation (10), and the balance between the two effects, controlled by α, is considered to determine the behavior of the MHA. The detailed structure of the architecture corresponding to (10) is illustrated in Figure 1(a).

In the following experiments, we will employ scaled dot-product attention according to the usual Transformer design and introduce the coefficient $\frac{1}{\sqrt{d_k}}$ in the argument of the softmax function.

4.2 Text Generation: GPT-2

To determine the impact of MHA on Transformer performance, we first trained GPT-2 Small(124M) and Medium(350M) [39] on text generation task and tested their performance. The dataset used was WikiText103 [33]. The following experiments in this paper were conducted using up to eight A100 GPUs. The detailed training settings are described in the supplemental material.

To fairly compare the effectiveness of MHA, we trained the GPT-2 architecture and an architecture in which the self-attention layers of GPT-2 are replaced by MHA in the same setting from scratch and compared their perplexity. Table 1 shows the results. The interest of this paper is not to create a SOTA model with detailed hyperparameter tuning, etc., but to see the robustness of the MHA effect, so α was simply set to 0.5 based on rough hyperparameter search.

As Table 1 shows, there is a clear improvement in perplexity in both the Small and Medium MHA models. Hopfield networks have often been experimented with in comparison to encoder Transformers [40], but our result shows that such comparisons is also useful for decoders.

Table 1: Comparison of the perplexity of GPT-2 and its MHA counterpart trained on the WikiText103 dataset for two cases: GPT-2 Small with 124M parameters and GPT-2 Medium with 350M parameters. In both cases, the introduction of MHA improved the perplexity.

Small(124M)		Medium(350M)	
self-attention	MHA($\alpha = 0.5$)	self-attention	MHA($\alpha = 0.5$)
22.87	**20.70**	20.85	**19.61**

4.3 Text Generation: LLaMA Architecture

To evaluate the effectiveness of Modern Hopfield Attention in more practical text generation architectures, we conducted additional experiments on LLaMA, in addition to GPT-2, using the miniLLaMA implementation. Furthermore, besides WikiText-103, we individually examined cases where CNN DailyMail [18] and BookCorpus [56] were used as training datasets. The results are summarized in Table 2. Even in practical architectures such as LLaMA, whose refined design aims to enhance performance, MHA was found to exert a consistent improvement in perplexity, demonstrating its systematic effectiveness beyond simpler baseline models.

dataset	self-attention	MHA
WikiText-103	14.49	**14.29**
DailyMail	19.36	**18.97**
BoocCorpus	23.76	**23.50**

Table 2: Comparison of the perplexity of LLaMA and its MHA counterpart ($\alpha = 0.5$) trained on various datasets. In all cases, the introduction of MHA led to improved perplexity.

4.4 Image Recognition: ViT

Next, the Vision Transformer (ViT) was employed as the Transformer decoder model, and again to fairly compare the effect of MHA, two architectures, the ViT architecture and the architecture in which the self-attention layers of ViT are replaced by MHA, were trained in the same configuration. We trained these models in image recognition tasks.

The model used in this study is ViT [12], and the data sets used are CIFAR10/CIFAR100 [29] and ImageNet-1k [8]. The detailed training setup is shown in the supplemental material.

model size	model type	CIFAR10	CIFAR100
ViT-Tiny(5.5M)	self-attention	93.265	**73.080**
	MHA($\alpha = 0.5$)	93.015	72.030
	MHA($\alpha = 0.7$)	**93.775**	72.570
ViT-Small(22M)	self-attention	**95.450**	74.485
	MHA($\alpha = 0.5$)	95.335	75.420
	MHA($\alpha = 0.7$)	95.440	**75.590**
ViT-Base(86M)	self-attention	96.190	75.360
	MHA($\alpha = 0.5$)	96.175	**76.215**
	MHA($\alpha = 0.7$)	**96.490**	75.590
ViT-Large(303M)	self-attention	96.310	72.910
	MHA($\alpha = 0.5$)	96.500	**75.775**
	MHA($\alpha = 0.7$)	**96.690**	75.365

Table 3: Experimental results are shown for ViTs and their MHA counterparts. For simple tasks such as CIFAR10, performance is close to saturation and there is no clear effect of MHA. On the other hand, for CIFAR100, the performance improvement due to MHA is clear for the larger model. This is a common property of $\alpha = 0.5$ and $\alpha = 0.7$.

4.4.1 CIFAR10/100

First, as a simple case, we review the results for CIFAR10 in the left column of the Table 3; for CIFAR10, the effect of MHA is not clearly visible, partly because the performance is basically close to saturation due to the ease of the task. However, it is interesting to note that the effect of MHA is starting to appear in the Base and Large models, which have a high learning capacity. In any case, CIFAR10 is not a sufficient task for the purpose of observing changes in Transformer performance with scratch training.

So let's look at the results for CIFAR100, where the task is more difficult: as shown in Table 3, the larger the model, the larger and clearer the improvement compared to the baseline ViT. Interestingly,

in both cases of the two α choices shown here, the performance improvement relative to ViT can be seen when the model is larger than the Small model.

4.4.2 ImageNet-1k

In the experiments on ImageNet-1k, due to computational resource constraints, we adopt ViT-B (86M) as a model of good enough size to obtain nontrivial training results. The results of 300-epoch training of ViT-B and its MHA counterpart from scratch with ImageNet-1k are shown in Table 4. Following the standard training setup, AdamW[32] was used for optimizer and cosine decay for learning rate scheduling. Random erasing[54], mixup[52], cutmix[51], and RandomAugment[6] were used for augmentation. For details, please refer to the supplemental material.

Table 4: Classification validation accuracies for ViT-B(86M) and its MHA counterpart.

Data set	self-attention	MHA($\alpha = 0.5$)	MHA($\alpha = 0.7$)
ImageNet-1k	76.074	76.434	**77.058**

As shown in Table 4, the performance improvement in ViT-B was also observed in ImageNet-1k. Although the performance improvement is less than 1%, this performance difference is considered a non-trivial result compared to the examples in previous studies on ViT improvement. As in previous experiments, this increase in performance is produced by adding only a small amount of computation without adding any training parameters. This is an interesting result, which suggests that hidden states may help improve attention mechanisms. In the next chapter, we will investigate both theoretically and experimentally regarding how hidden states produce these performance gains.

4.4.3 Downstream Tasks

To evaluate MHA's effectiveness across diverse tasks, we measured the transfer performance of a pre-trained ImageNet model using linear probing. Using a pre-trained ViT and its MHA counterpart as backbones, we conducted transfer learning experiments on four commonly used downstream datasets (Oxford Flowers 102 [35], Food-101 [4], Stanford Dogs [27], and Stanford Cars [28]). Results in Table 5 show that the MHA variants achieve consistently good transfer performance.

dataset	self-attention	MHA
Flower102	81.15	**93.85**
Food101	74.51	**87.99**
Stanford dogs	**95.00**	83.64
Stanford cars	51.54	**87.54**

Table 5: Comparison of the transfer accuracy of ImageNet-1K pre-trained ViT and its MHA counterpart ($\alpha = 0.7$) on various downstream datasets.

4.5 Effect of Combining α and α'

Our update rule (10) has two hyperparameters α and α', but for simplicity, we have so far restricted our discussion to the case where both values are equal $\alpha = \alpha'$. However, as shown in Figure 1(a), these two quantities essentially work differently. α is a quantity that balances the value after attention computation and the strength of the skip connections in the attention module. On the other hand, α' is the coefficient of the exponential moving average in accumulating the attention scores to hidden states.

A nontrivial result (10) derived from MCHN is that these two independent effects are simultaneously added to the Transformer. To see whether these two are really both necessary, or whether they work in concert, let us try an experiment in which α and α' are varied independently.

Table 6 shows the change in performance when one of the hyperparameters is fixed at 0.5 and the value of the other is varied. When only α is moved from the original $\alpha = 0.5 = \alpha'$ to $\alpha = 1$, the performance drops to chance-level accuracy. This is evident from the fact that all values except for the skip connection are set to 0. On the other hand, when α is set to 0, the performance degrades to

Table 6: Performance change of ViT-T when two hyperparameters are changed independently

CAIFAR100 MHA($\alpha = 0.5$)											
α'	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
score	71.16	71.13	72.29	70.77	72.12	72.13	72.06	72.02	71.98	70.72	66.10
CAIFAR100 MHA($\alpha' = 0.5$)											
α	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
score	69.89	71.20	71.26	71.64	72.02	72.13	72.66	70.52	70.46	67.70	1.00

69.89. Thus, it can be seen that further performance improvement is realized by adding not only α' but also α. Similarly, when α is fixed, setting α' to 0 also results in poorer performance.

5 How MHA improves Transformers

5.1 Problem and Improvement of Transformer Layers

Next, let us examine why MHA leads to performance gains in various tasks. It is known that as the depth of Transformer increases, training becomes more difficult and performance tends to saturate rapidly. The phenomenon of rank collapse has been discussed as one cause of this problem. It is possible that our model mitigates the problem without explicit regularization or other means. Therefore, we provide below some theoretical and empirical results that support this hypothesis.

5.1.1 Rank Collapse

It has been observed that as the depth of the Vision Transformer increases, the patch tokens become extremely similar and rapidly lose diversity [44][55]. This phenomenon is now understood as token uniformity, rank collapse, or oversmoothing [11, 44, 55, 50, 36, 43, 15, 14, 37, 47, 2, 13][2]. Various innovations have been proposed to reduce this problematic phenomenon in order to improve Transformer performance [44, 55, 2].

Rank collapse [11] is defined as a phenomenon in which Transformer feature rapidly collapses into a rank 1 matrix with increasing depth. Thus, for the feature $\boldsymbol{X}^{(L)}$ of the L-th layer, rank collapse is formulated as the property that the residual of deep Transformer feature rapidly converges to zero as follows

$$\|\mathrm{Res}(\boldsymbol{X}^{(L)})\| \approx 0 \text{ for } L \gg 1, \tag{11}$$

where $\mathrm{Res}(\boldsymbol{X})$ is the residual $\mathrm{Res}(\boldsymbol{X}) = \boldsymbol{X} - \mathbf{1}\boldsymbol{x}^\top$ for $\boldsymbol{x} = \arg\min_{\boldsymbol{x}} \|\boldsymbol{X} - \mathbf{1}\boldsymbol{x}^\top\|$. This convergence means the feature is approximately a rank one matrix $\boldsymbol{X}^{(L)} \approx \mathbf{1}\boldsymbol{x}^\top$.

5.2 Theoretical Implication

For a clear theoretical analysis of the causes of rank collapse, we consider a deep network consisting of only the self-attention layers according to [11]. It is also straightforward to extend the discussion to the actual Transformer architecture [11]. Let us consider a self-attention-only network consisting of L layers without skip connection

$$\mathrm{AttnNet}(\boldsymbol{X}) = \mathrm{MHSA} \circ \cdots \circ \mathrm{MHSA}(\boldsymbol{X}). \tag{12}$$

$\mathrm{MHSA}(\boldsymbol{X})$ is the multi-head self-attention module. The number of heads and embedding dimension of each MHSA are H and d_k. In [11], this attention-only network has been shown to cause very serious rank collapse:

Theorem 5.1 ([11]). *The norm of the residual of attention-only network AttnNet($\boldsymbol{X}$) decays as*

$$\|\mathit{Res}\left(\mathit{AttnNet}(\boldsymbol{X})\right)\|_{1,\infty} \leq (rC)^{\frac{3^L-1}{2}} \|\mathit{Res}\left(\boldsymbol{X}\right)\|_{1,\infty}^{3^L}, \tag{13}$$

where $r = \frac{8H}{\sqrt{d_k}}$ and C is certain constant. This suggests the double exponential decay of the rank.

[2]The rank collapse in [11] refers to the phenomenon where the tokens corresponding to each row of a feature become perfectly proportional vectors. This means perfect token uniformity. On the other hand, the phenomenon observed in actual Transformers is that many, if not all, tokens are perfectly aligned, forming a group of tokens with a mutual cosine similarity of 1.

The definition of the norm $\|\cdot\|_{1,\infty}$ in this paper is the composite of operator norms $\|\boldsymbol{X}\|_{1,\infty} = \sqrt{\|\boldsymbol{X}\|_1 \|\boldsymbol{X}\|_\infty}$.

In [11], it was shown that skip connection and the addition of an FFN layer are effective in reducing this serious collapse. Interestingly, however, even though our MHA is not specifically designed to prevent rank collapse, it is able to prevent the decay phenomenon in attention-only networks without any skip connection. Even when removing skip connections completely by setting $\alpha = 0$, a non-zero α' leads to the following mitigation of rank collapse in the attention-only network:

Theorem 5.2. *By keeping non-zero α', the upper-bound of inequality evaluation is improved as follows*

$$\|\mathit{Res}\big(\mathit{AttnNet}(\boldsymbol{X})\big)\|_{1,\infty} \leq \max{}_{m=0}^{L} \left(r(1-\alpha')C_1\right)^{\frac{3^m-1}{2}} \left(r\alpha' C_2\right)^{3^m(L-m)} \|\mathit{Res}\big(\boldsymbol{X}\big)\|_{1,\infty}^{3^m}. \quad (14)$$

This suggests the avoidance of exponential decay.

Proof. See the supplemental material for detailed proof. The sketch of the proof is as follows: by introducing the hidden state as $\alpha' \neq 0$, the decaying effect of rank by single attention layer can be evaluated as follows

$$\|\mathrm{Res}\big(\mathrm{MHSA}(\boldsymbol{X}\big)\| \leq \max\left(r_1(1-\alpha')\|\mathrm{Res}\big(\boldsymbol{X}\big)\|^3,\, r_2\alpha'\|\mathrm{Res}\big(\boldsymbol{X}\big)\|\right), \quad (15)$$

where $r_{1,2} = rC_{1,2}$ and the norm here is $\|\cdot\|_{1,\infty}$. Notice that the second argument in the max function significantly reduces the third-order decaying effect in [11]. By applying this inequality repeatedly over L layers, we obtain the following inequality

$$\|\mathrm{Res}\big(\mathrm{AttnNet}(\boldsymbol{X})\big)\| \leq \max\left(\left(r_1(1-\alpha')\right)^{\frac{3^L-1}{2}} \|\mathrm{Res}\big(\boldsymbol{X}\big)\|^{3^L}, \cdots, \left(r_2\alpha'\right)^L \|\mathrm{Res}\big(\boldsymbol{X}\big)\|\right), \quad (16)$$

where $\cdots$ means $(r_1(1-\alpha'))^{(3^m-1)/2}(r_2\alpha')^{3^m(L-m)}(\|\mathrm{Res}(\boldsymbol{X})\|)^{3^m}$ for $m = 1, \cdots, L-1$. □

On the right hand side of this inequality (14), the $m = L$ term is the very term that created the double exponential decay of the original self-attention mechanism [11], but the $m = 0$ term dominates in (14) and relaxes the rank decay to linear decay as $\left(r\alpha' C_2\right)^L \|\mathrm{Res}\big(\boldsymbol{X}\big)\|_{1,\infty}$ since

$$(r_1(1-\alpha'))^{(3^m-1)/2}(r_2\alpha')^{3^m(L-m)}(\|\mathrm{Res}(\boldsymbol{X})\|)^{3^m} < (r_2\alpha')^L(\|\mathrm{Res}(\boldsymbol{X})\|)^L. \quad (17)$$

Note that we assume $r_{1,2}, \|\mathrm{Res}(\boldsymbol{X})\| < 1$ following the logic of [11]. This decaying factor is controlled by the hidden states of the h-th head of the ℓ-th layer $\boldsymbol{H}_{\ell,h} = \alpha'\boldsymbol{H}_{\ell-1,h} + (1-\alpha')\boldsymbol{Q}_{\ell,h}\boldsymbol{K}_{\ell,h}^\top$ and the weight matrix $\boldsymbol{W}_{VO,h}^{(\ell)}$ for the value and output linear projection of attention module as $C_2 = \max_\ell \max_h \|\boldsymbol{W}_{VO,h}^{(\ell)}\|_{1,\infty}\|\boldsymbol{H}_{\ell,h}\|_1$.

In [11], such an effect was created by introducing skip connection, but in the MHA, the hidden state contribution already produces such an effect without using skip connection. Also, setting $\alpha' = 0$ reproduces the double exponential decay results of the original attention-only network (13).

5.3 Empirical Results

Using the theoretical analysis setup used in previous studies, we showed that MHA can effectively prevent rank collapse in the previous section. However, since these setups are based on several theoretical simplifications, it is unclear whether the rank collapse reduction also occurs in actual Transformers. In particular, it is not clear whether the introduction of MHA has any further effect in usual architectures with skip connection to reduce rank collapse. In this section, we will confirm that MHA does indeed further reduce rank collapse in a few controlled experiments.

Since the skip-free network was shown to suffer from rank collapse as it gets deeper, let's examine the effect of MHA on the actual performance degradation with depth. Table 7 shows the results of trained models from depths 1 to 12 for the skip-free networks and their MHA versions, and evaluating their performance. As can be seen from the results in the left column of the table, when the depth increases beyond 4 layers, the performance drops sharply due to multilayering. On the other hand, for the models in the right column using MHA, it can be seen that the degradation of the model due to multilayering is kept at a fairly mild level. Thus, the MHA model can effectively utilize the depth of

Table 7: Changes in performance as skip-free networks based on ViT-T are deepened.

depth	self-attention		MHA	
	CIFAR10	CIFAR100	CIFAR10	CIFAR100
1	55.08	30.90	65.41	40.08
2	63.72 ↑	40.06 ↑	79.75 ↑	56.94 ↑
4	57.38 ↓	32.25 ↓	85.74 ↑	64.39 ↑
8	48.59 ↓	17.19 ↓	80.34 ↓	49.90 ↓
12	10.00 ↓	1.00 ↓	10.00 ↓	1.00 ↓

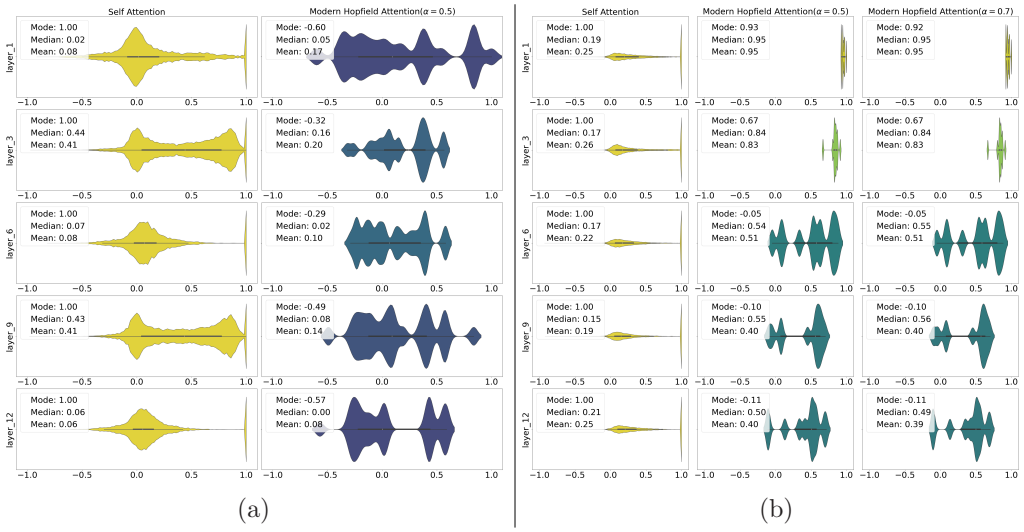


Figure 2: The violin plots of cosine similarity between tokens in several layers for (a) GPT-2 (Medium) trained on Wikitext103 and (b) ViT-B trained on CIFAR100. MHA layers with high average similarity of tokens exist, but tokens with a perfect similarity of 1, as in the case of self-attention, disappear, preventing their ranks from dropping.

the model than the original model. Therefore, it is highly likely that MHA can significantly improve rank collapse, which becomes more severe as the network becomes deeper, even in real networks.

Next, let us examine cases of actual Transformer architectures with skip connections and FFN layers. Figure 2 shows the measured cosine similarity between tokens in several layers for GPT-2 (Medium) and ViT-B trained on CIFAR100, displayed as violin plots. See the supplementary material for more detailed plots. It is noteworthy that in the cases of normal GPT-2 and ViT-B, the mode of similarity is 1.0 for all layers, while the violin plot shows a sharp peak around 1.0. This indicates that even with the addition of the skip connection and FFN layers, there is still a non-negligible token uniformity, or partial rank collapse. On the other hand, the results for the GPT-2 and ViT models with MHA show that the peaks in the original models have disappeared and the mode values have been reduced to very small values. This indicates that MHA does indeed play a role in dramatically removing token uniformity in GPT-2 and ViT.

6 Conclusion

In this paper, we examine the question of whether new insights can be obtained from the modern Hopfield network for Transformer. The results showed that by introducing the hidden state of MCHN into Transformer, a new attention mechanism called MHA, which inherits attention scores from layer to layer, has been discovered and can be useful for improving ViT and GPT performance. MHA was also found to play a role in solving the rank collapse problem in deep Transformer. The MHA's mechanism to prevent the rank collapse may have contributed to Transformer's improved performance. We hope that this research will open new possibilities for the systematic design of Transformer architectures using Hopfield networks.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract reflects the results of Sections 5 and 2.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The Supplementary Material describes the limitations.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: The Supplementary Material describes proof.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Codes and settings are included in the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code and other materials included in the supplemental material will be released on GitHub after acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: settings and details are included in the Supplementary Material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [NA]

   Justification: Due to the high cost of training Transformer models.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Computational resources are described in 4.2

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The study was conducted in compliance with the code of ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This study is a basic study of the Hopfield network and Transformer, and therefore does not directly have a broader impact.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Since this is basic research, such risks are minimal.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: A link to the referenced code is posted at the top of the code.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

References

[1] S-I Amari. Learning patterns and pattern sequences by self-organizing nets of threshold elements. *IEEE Transactions on computers*, 100(11):1197–1206, 1972.

[2] Jiawang Bai, Li Yuan, Shu-Tao Xia, Shuicheng Yan, Zhifeng Li, and Wei Liu. Improving vision transformers by revisiting high-frequency components. In *European conference on computer vision*, pages 1–18. Springer, 2022.

[3] Alberto Bietti, Vivien Cabannes, Diane Bouchacourt, Herve Jegou, and Leon Bottou. Birth of a transformer: A memory viewpoint. *Advances in Neural Information Processing Systems*, 36:1560–1588, 2023.

[4] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014.

[5] Thomas F Burns and Tomoki Fukai. Simplicial hopfield networks. In *The Eleventh International Conference on Learning Representations*.

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[7] Mete Demircigil, Judith Heusel, Matthias Löwe, Sven Upgang, and Franck Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168:288–299, 2017.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[10] Anxhelo Diko, Danilo Avola, Marco Cascio, and Luigi Cinque. Revit: Enhancing vision transformers feature diversity with attention residual connections. *Pattern Recognition*, 156:110853, 2024.

[11] Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International Conference on Machine Learning*, pages 2793–2803. PMLR, 2021.

[12] 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. In *International Conference on Learning Representations*, 2021.

[13] Gbètondji JS Dovonon, Michael M Bronstein, and Matt J Kusner. Setting the record straight on transformer oversmoothing. *arXiv preprint arXiv:2401.04301*, 2024.

[14] Xiaojun Guo, Yifei Wang, Tianqi Du, and Yisen Wang. Contranorm: A contrastive learning perspective on oversmoothing and beyond. *arXiv preprint arXiv:2303.06562*, 2023.

[15] Bobby He, James Martens, Guodong Zhang, Aleksandar Botev, Andrew Brock, Samuel L Smith, and Yee Whye Teh. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. *arXiv preprint arXiv:2302.10322*, 2023.

[16] Ruining He, Anirudh Ravula, Bhargav Kanagal, and Joshua Ainslie. Realformer: Transformer likes residual attention. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 929–943, 2021.

[17] Zexue He, Leonid Karlinsky, Donghyun Kim, Julian McAuley, Dmitry Krotov, and Rogerio Feris. Camelot: Towards large language models with training-free consolidated associative memory. In *First Workshop on Long-Context Foundation Models@ ICML 2024*.

[18] Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

[19] Claus Hofmann, Simon Lucas Schmid, Bernhard Lehner, Daniel Klotz, and Sepp Hochreiter. Energy-based hopfield boosting for out-of-distribution detection. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

[20] Benjamin Hoover, Yuchen Liang, Bao Pham, Rameswar Panda, Hendrik Strobelt, Duen Horng Chau, Mohammed Zaki, and Dmitry Krotov. Energy transformer. *Advances in neural information processing systems*, 36:27532–27559, 2023.

[21] Benjamin Hoover, Hendrik Strobelt, Dmitry Krotov, Judy Hoffman, Zsolt Kira, and Polo Chau. Memory in plain sight: A survey of the uncanny resemblances between diffusion models and associative memories. In *Annual Conference on Neural Information Processing Systems*, 2023.

[22] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[23] Jerry Yao-Chieh Hu, Pei-Hsuan Chang, Haozheng Luo, Hong-Yu Chen, Weijian Li, Wei-Po Wang, and Han Liu. Outlier-efficient hopfield layers for large transformer-based models. In *International Conference on Machine Learning*, pages 19123–19152. PMLR, 2024.

[24] Jerry Yao-Chieh Hu, Thomas Lin, Zhao Song, and Han Liu. On computational limits of modern hopfield models: a fine-grained complexity analysis. In *Proceedings of the 41st International Conference on Machine Learning*, pages 19327–19343, 2024.

[25] Jerry Yao-Chieh Hu, Donglin Yang, Dennis Wu, Chenwei Xu, Bo-Yu Chen, and Han Liu. On sparse modern hopfield model. *Advances in Neural Information Processing Systems*, 36:27594–27608, 2023.

[26] Georgios Iatropoulos, Johanni Brea, and Wulfram Gerstner. Kernel memory networks: A unifying framework for memory modeling. *Advances in neural information processing systems*, 35:35326–35338, 2022.

[27] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, and Fei-Fei Li. Novel dataset for fine-grained image categorization: Stanford dogs. In *Proc. CVPR workshop on fine-grained visual categorization (FGVC)*, volume 2, 2011.

[28] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 554–561, 2013.

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[30] Dmitry Krotov and John J Hopfield. Dense associative memory for pattern recognition. *Advances in neural information processing systems*, 29, 2016.

[31] Dmitry Krotov and John J Hopfield. Large associative memory problem in neurobiology and machine learning. In *International Conference on Learning Representations*, 2021.

[32] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[33] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

[34] Beren Millidge, Tommaso Salvatori, Yuhang Song, Thomas Lukasiewicz, and Rafal Bogacz. Universal hopfield networks: A general framework for single-shot associative memory models. In *International Conference on Machine Learning*, pages 15561–15583. PMLR, 2022.

[35] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008.

[36] Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. *Advances in Neural Information Processing Systems*, 35:27198–27211, 2022.

[37] Namuk Park and Songkuk Kim. How do vision transformers work? *arXiv preprint arXiv:2202.06709*, 2022.

[38] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[39] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

[40] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, Philipp Seidl, Michael Widrich, Lukas Gruber, Markus Holzleitner, Thomas Adler, David Kreil, Michael K Kopp, et al. Hopfield networks is all you need. In *International Conference on Learning Representations*, 2021.

[41] Bishwajit Saha, Dmitry Krotov, Mohammed J Zaki, and Parikshit Ram. End-to-end differentiable clustering with associative memories. In *International Conference on Machine Learning*, pages 29649–29670. PMLR, 2023.

[42] Tommaso Salvatori, Yuhang Song, Yujian Hong, Lei Sha, Simon Frieder, Zhenghua Xu, Rafal Bogacz, and Thomas Lukasiewicz. Associative memories via predictive coding. *Advances in neural information processing systems*, 34:3874–3886, 2021.

[43] Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen Lee, and James T Kwok. Revisiting over-smoothing in bert from the perspective of graph. *arXiv preprint arXiv:2202.08625*, 2022.

[44] Yehui Tang, Kai Han, Chang Xu, An Xiao, Yiping Deng, Chao Xu, and Yunhe Wang. Augmented shortcuts for vision transformers. *Advances in Neural Information Processing Systems*, 34:15316–15327, 2021.

[45] Danil Tyulmankov, Ching Fang, Annapurna Vadaparty, and Guangyu Robert Yang. Biological learning in key-value memory networks. *Advances in Neural Information Processing Systems*, 34:22247–22258, 2021.

[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[47] Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*, 2022.

[48] James CR Whittington, Joseph Warren, and Tim EJ Behrens. Relating transformers to models and neural representations of the hippocampal formation. In *International Conference on Learning Representations*.

[49] Dennis Wu, Jerry Yao-Chieh Hu, Teng-Yun Hsiao, and Han Liu. Uniform memory retrieval with larger capacity for modern hopfield models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 53471–53514, 2024.

[50] Hanqi Yan, Lin Gui, Wenjie Li, and Yulan He. Addressing token uniformity in transformers via singular value transformation. In *Uncertainty in artificial intelligence*, pages 2181–2191. PMLR, 2022.

[51] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[52] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

[53] Jinsong Zhang, Qiang Fu, Xu Chen, Lun Du, Zelin Li, Gang Wang, Shi Han, Dongmei Zhang, et al. Out-of-distribution detection based on in-distribution data patterns memorization with modern hopfield energy. In *The Eleventh International Conference on Learning Representations*, 2022.

[54] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[55] Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021.

[56] Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27, 2015.