Associative Transformer

Yuwei Sun[1,2], Hideya Ochiai[3], Zhirong Wu[4], Stephen Lin[4*], Ryota Kanai[1*]

[1]Araya Research, [2]RIKEN AIP, [3]The University of Tokyo, [4]Microsoft Research

Abstract

Emerging from the pairwise attention in conventional Transformers, there is a growing interest in sparse attention mechanisms that align more closely with localized, contextual learning in the biological brain. Existing studies such as the Coordination method employ iterative cross-attention mechanisms with a bottleneck to enable the sparse association of inputs. However, these methods are parameter inefficient and fail in more complex relational reasoning tasks. To this end, we propose Associative Transformer (AiT) to enhance the association among sparsely attended input tokens, improving parameter efficiency and performance in various vision tasks such as classification and relational reasoning. AiT leverages a learnable explicit memory comprising specialized priors that guide bottleneck attentions to facilitate the extraction of diverse localized tokens. Moreover, AiT employs an associative memory-based token reconstruction using a Hopfield energy function. The extensive empirical experiments demonstrate that AiT requires significantly fewer parameters and attention layers outperforming a broad range of sparse Transformer models. Additionally, AiT outperforms the SOTA sparse Transformer models including the Coordination method on the Sort-of-CLEVR dataset.

1. Introduction

Transformer models use pairwise attention to establish correlations among disparate segments of input information [1, 2]. Emerging from the pair-wise attention, there is a growing interest in leveraging sparse interactions for localized, contextual learning. This sparsity attribute has demonstrated advantages in enhancing learning performance and efficiency [3, 4]. Sparse knowledge association can find resonance with the neuroscientific grounding of the Global Workspace Theory (GWT) [5–8]. GWT explains a fundamental cognitive architecture for working memory where diverse specialized modules compete to write information into a shared workspace through a communication bottle-

neck. The bottleneck facilitates the processing of content-addressable information using attention guided by contents in the shared workspace [9, 10].

A bottleneck guides models to generalize in a manner consistent with the underlying data distribution through inductive biases of sparsity [11, 12], resulting in superior performance. In this regard, the Coordination method [13] is an initial attempt to assess the effectiveness of such a bottleneck in Transformers. Unfortunately, its design is parameter inefficient and often fails in more complex relational reasoning tasks [14–20]. In this work, we aim to enhance sparse attention capability of Transformers by leveraging low-rank explicit memory to learn a diverse set of priors to guide the attention. Furthermore, drawing inspiration from associative memory based on Hebbian learning, we utilize continuous Hopfield networks [21, 22] to reconstruct input tokens from the explicit memory. The associative memory enhances the effective association of sparsely attended historical inputs.

To this end, we propose Associative Transformer (AiT), a sparse representation learner. AiT learns explicit memory for each attention layer to facilitate the extraction of localized features at different abstraction levels. The explicit memory evolves into specialized priors through the bottleneck attention, with each focusing on a specific spatial relation of input tokens. We further propose a bottleneck attention balance loss to encourage diversity among the learned priors. Furthermore, the acquired priors function as distinct attractors within the associative memory of Hopfield networks, facilitating token reconstruction from memory. Extensive empirical results demonstrate that the emerging specialization of priors and the reconstruction within associative memory in AiT could significantly enhance parameter efficiency and model performance across a broad spectrum of tasks.

Overall, our main contributions are three-fold:

1) We propose the Associative Transformer (AiT) to tackle inefficient sparse attention mechanisms in conventional Transformers, leveraging emerging specialized priors for guided bottleneck attention and token reconstruction within associative memory (Section 3.3).

2) The learned priors function as attractors of Hopfield

---

*[*]Both authors contributed equally to the supervision of this work.
Corresponding author: Yuwei Sun (yuwei_sun@araya.org).

networks, facilitating information correlation and retrieval from historical inputs. To the best of our knowledge, this is the first study on consolidating Hopfield networks with the sparse attention mechanism (Section 3.4).

3) Extensive experiments including an ablation study demonstrate enhanced parameter efficiency of AiT in various classification and relational reasoning tasks, outperforming the Coordination method (Section 4).

2. Related work

This section provides a summary of the most relevant recent work on sparse attention, latent memory-enabled Transformers, and external memory mechanisms. We investigate the relatedness of these studies to various properties of the Global Workspace Theory (GWT) in Appendix B.

Sparse Transformers Studies on sparse Transformers have explored consolidating latent memory to extract localized, contextual representations from inputs [13, 23–27]. For example, Perceiver [26, 27] utilized iterative cross-attention with a latent array and transformation to capture dependencies across input tokens. Set Transformer [23] and Linear Unified Nested Attention [25] also employed iterative cross-attention but without the latent transformation. Other methods, like Blockwise Self-Attention, relied on a strong assumption of predefined modularization for attention specialization [28]. In contrast, our method learns such modularization and attention specialization through end-to-end training. Unlike the vast majority of methods that employ latent memory, our proposed method leverages both explicit memory and associative memory for enhanced contextual learning capability.

In a similar approach to constructing sparse Transformers based on GWT, the Coordination method [13] introduced a bottleneck into cross-attention mechanisms to encourage neural module specialization. The results indicated that the competition via the bottleneck could contribute to a small set of specialized priors in latent memory to facilitate relational learning. However, the number of priors was limited to fewer than 10, all with the same dimension as the input representation. By contrary, our method employs low-rank explicit memory to learn a significantly richer set of priors (up to 128 priors learned from a pool of 32.8k tokens). Moreover, the Coordination method relies on iterative cross-attentions, while our work focuses on novel associative memory-enabled sparse attention.

This work is also related to modular neural networks and mixture of experts [29, 30] in terms of competition in the shared workspace. Separating the information processing within Transformers into distinct components, depending on the input, has shown more flexibility in data processing.

Memory mechanisms External memory such as tape storage and associative memory, has been employed in deep neural networks [31–33], with recent studies exploring the potential use of Hopfield networks and their modern variants [21, 22, 34]. In this work, we incorporate Hopfield networks as an integral element in the sparse attention mechanism of the proposed Associative Transformer. This approach is fundamentally different from the previous studies such as Energy Transformer [35] that mainly focused on utilizing Hopfield networks independently of the attention mechanism.

3. Associative Transformer

This section discusses the key technical underpinnings of the Associative Transformer (AiT), focusing on the newly devised *Global Workspace Layer (GWL)*. The GWL integrates three main elements: low-rank explicit memory, bottleneck attention, and associative memory (Figure 1). It facilitates efficient writing and reading in explicit memory by promoting competition among input tokens through a specialized top-k bottleneck attention mechanism that limits cross-attention capacity. Two techniques are employed to maintain balanced information writing: Exponential Weighted Moving Average (EWMA), which stabilizes memory updates by combining old and new information, and a novel bottleneck attention balance loss, which encourages diverse token selection and counteracts the loss of specificity across layers. After writing, the explicit memory contents are broadcast back to the input tokens for retrieval within a Hopfield network-based associative memory, reconstructing the tokens into more globally meaningful representations that serve as the output of the GWL. In practice, the GWL can be integrated with conventional attention layers to enhance the efficiency of Transformer models, boosting their ability to learn abstract spatial relations across samples and time steps.

3.1. Vision Transformers for classification tasks

Vision Transformers (ViT) tackle image classification tasks by processing sequences of image patches. The pre-processing layer partitions an image into non-overlapping patches. Let $x \in \mathbb{R}^{H \times W \times C}$ be an input from the dataset X, where (H, W) is the resolution of the image and C is the number of channels. x is separated into a sequence of patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$, where (P, P) is the resolution of each image patch and $N = \frac{HW}{P^2}$ is the number of patches. These patches are mapped to embeddings $v_p \in \mathbb{R}^{N \times E}$ with the linear projection. ViT leverages self-attention to map a query and a set of key-value pairs to an output. The patch embeddings are used to obtain the query, key, and value based on linear transformations $W^Q \in \mathbb{R}^{E \times D}$, $W^K \in \mathbb{R}^{E \times D}$, and, $W^V \in \mathbb{R}^{E \times D}$. The output of an individual attention head is a weighted sum of

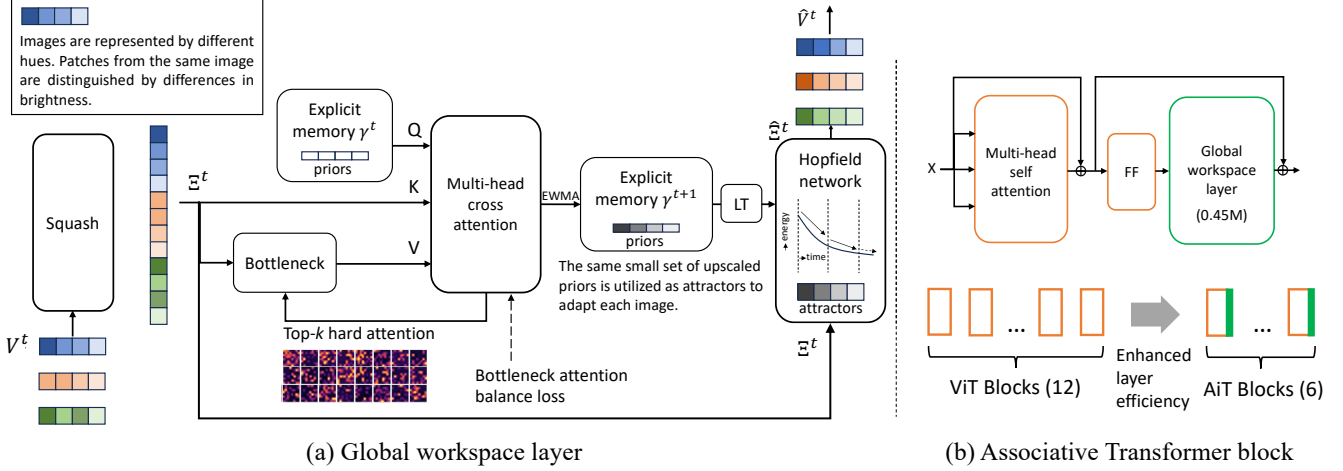(a) Global workspace layer   (b) Associative Transformer block

Figure 1. The scheme of the Associative Transformer (AiT). (a) In a global workspace layer, the input $\mathbb{R}^{B \times N \times E}$ is squashed into vectors $\mathbb{R}^{BN \times E}$. The squashed representations are projected to a latent space of dimension $D << E$ and are sparsely selected to update the explicit memory via a fixed bottleneck $k << BN$. The Hopfield network utilizes the memory to reconstruct the input tokens, where a learnable linear transformation (LT) scales the memory contents back to the input dimension E. (b) The Associative Transformer block consists of self attention, feed-forward layers, and the global workspace layer. Compared to Vision Transformer (ViT), leveraging the global workspace layer enhances the layer efficiency. A shallower 6-layer AiT is shown to outperform a 12-layer ViT (see Table 1).

the values, $\text{softmax}(\frac{W_i^Q v (W_i^K v)^T}{\sqrt{D}}) W_i^V v$.

We specifically consider supervised learning with F categories. Let $y \in \{1, 2, ..., F\} = Y$ denote a label. Given a Transformer model $f(x)$, the prediction is given by $\hat{y} = \arg\max_j f(x)_j$ where $f(x)_j$ denotes the jth element of $f(x)$. The training is attained by minimizing the loss with respect to the model parameter θ:

$$\ell_{\text{class}}(\theta) = J(Y, f(X; \theta)), \quad (1)$$

where J denotes the cross entropy loss.

3.2. Low-rank priors in explicit memory

To induce memory in large-scale models like Transformers, we focus on learning more efficient explicit memory, which is later consolidated with Hopfield networks. This allows us to decouple the memory storage from the Hopfield network learning, such that the memory could be gradually updated with old and new memory using methods such as exponentially weighted moving average (EWMA). The explicit memory is only trainable during model training and frozen during evaluation. The proposed Global Workspace Layer enables efficient writing and reading in the explicit memory to learn a set of low-rank priors through a tailored bottleneck attention mechanism.

In particular, we employ a squash layer to concatenate patch representations within the entire batch $V \in \mathbb{R}^{B \times N \times E}$ into vectors $\Xi \in \mathbb{R}^{BN \times E}$, fostering association among tokens across samples. In practice, the number of tokens $B \times N$ varies depending on the batch size B. The attention

bottleneck with a fixed capacity k restricts the number of attended tokens to a constant value. The output of the squash layer is then attended to by the multi-head cross-attention using the contents of the explicit memory as queries. The explicit memory is comprised of M learnable low-rank priors $\gamma = \mathbb{R}^{M \times D}$ where D is the dimension of the prior, $D << E$. Priors are general assumptions about tokens to guide the attention for the token selection. The priors are utilized as various queries to compute cross-attentions that extract various sets of tokens and update the explicit memory. We found that a rank of 32 and initialization of the priors with a Gaussian distribution works well in most cases.

We devise a tailored cross-attention mechanism to update the explicit memory using the squash layer's output $\Xi \in \mathbb{R}^{BN \times E}$. Notably, in the cross-attention, the query is a function of the current memory content $\gamma = \{\gamma_i\}_{i=1}^M$. The key and value are functions of the squash layer's output Ξ. The attention scores A_i for head i are computed as follows $A_i(\gamma, \Xi) = \text{softmax}(\frac{\gamma(\Xi W_i^K)^T}{\sqrt{D}})$. This is the case of soft attention without constraints on the bottleneck capacity.

3.3. Bottleneck attention with a limited capacity

We discuss the case of limiting the cross-attention capacity via top-k bottleneck attention. The bottleneck attention constrains the number of tokens that can be attended to, fostering competition among tokens and ensuring the selection of the most relevant tokens. Notably, we select the tokens with the top-k highest attention scores from A_i, $h_i = \text{top-}k(A_i)\Xi W^V$. Additionally, to ensure a stable update of the explicit memory across different time steps, we

employ layer normalization and the exponentially weighted moving average (EWMA) strategies as follows:

$$\hat{\gamma}^{t+1} = (1 - \alpha) \cdot \gamma^t + \alpha \cdot \text{LN}(\text{Concat}(h_1, \ldots, h_S)W^O),$$

$$\gamma^{t+1} = \frac{\hat{\gamma}^{t+1}}{\sqrt{\sum_{j=1}^{M}(\hat{\gamma}_j^{t+1})^2}}, \tag{2}$$

where t is the batch time step, LN is the layer normalization, S is the number of attention heads, and α is a smoothing factor determining the decay rate of older observations, which is a small value such as 0.1. EWMA ensures stable memory updates by accumulating old and new memories. We found that different batch sizes B have little influence on the test performance. This can be attributed to the small decay rate that results in the stable memory update.

Bottleneck attention balance loss The sparsity induced by the bottleneck attention results in the emergence of specialized priors by attending to various tokens. However, cascading multiple bottleneck attentions across Transformer layers could lead to difficulty in efficiently forming specialized priors. As information flows through multiple bottleneck attention layers, the representations become diluted, necessitating a mechanism to counteract this inherent loss of input specificity for the effective selection of meaningful tokens across layers. To overcome this challenge, we propose the *bottleneck attention balance loss* to encourage a more diverse token selection with the bottleneck attention. The bottleneck attention balance loss $\ell_{\text{bottleneck}}$ comprises two components, i.e., the cumulative attention loss $\ell_{\text{importance}}$, which computes the total assigned attention scores for each token, and the selected instance loss ℓ_{loads}, which provides an estimate of the frequency of each token being selected. Then, the loss is computed for each token position $o \in \{1, 2, \ldots, B \times N\}$ in the squash layer output. We devise the bottleneck attention balance loss for attention head i by measuring the normalized variances of the cumulative attention loss and the selected instance loss across token positions as follows:

$$\ell_{\text{importance}_{i,o}} = \sum_{j=1}^{M} A_{i,j,o}, \; \ell_{\text{loads}_{i,o}} = \sum_{j=1}^{M}(A_{i,j,o} > 0), \tag{3}$$

$$\ell_{\text{bottleneck}_i} = \frac{\text{Var}(\{\ell_{\text{importance}_{i,o}}\}_{o=1}^{B \times N})}{(\frac{1}{B \times N}\sum_{o=1}^{B \times N}\ell_{\text{importance}_{i,o}})^2 + \epsilon} + \\ \frac{\text{Var}(\{\ell_{\text{loads}_{i,o}}\}_{o=1}^{B \times N})}{(\frac{1}{B \times N}\sum_{o=1}^{B \times N}\ell_{\text{loads}_{i,o}})^2 + \epsilon}, \tag{4}$$

where $A_{i,j,o}$ is head i's attention score for the jth prior at token position o, $\text{Var}(\cdot)$ denotes the variance, and ϵ is a small value to avoid division by zero.

The losses for the different heads are summed up and added to the classification loss, $\ell = \ell_{\text{class}} + \sigma \cdot \sum_{i=1}^{S} \ell_{\text{bottleneck}_i}$, where σ is a coefficient to adjust the ratio between the classification loss and the bottleneck attention balance loss.

3.4. Information retrieval with associative memory

After writing information from the input tokens into the explicit memory, the learned priors serve as attractors for a novel information retrieval component based on associative memory. The aim is to reconstruct the current input tokens towards more globally meaningful representations learned and stored in the explicit memory for enhanced token representations. The proposed architecture is an attractor network where a token converges to one of these attractors derived from the priors stored in the explicit memory, by decreasing its energy with respect to the explicit memory.

Attractors The learned priors in the explicit memory function as attractors within associative memory of a Hopfield network. Attractors have basins of attraction, and any input that enters an attractor's basin converges to that specific attractor. Attractors usually have the same dimension as input states. Since we employ low-rank priors $\gamma^{t+1} \in \mathbb{R}^{M \times D}$, a learnable linear transformation $f_{\text{LT}}(\cdot)$ projects the priors into the same dimension E as the input tokens before utilizing them as attractors.

Energy-based retrieval The upscaled priors $f_{\text{LT}}(\gamma^{t+1})$ are stored within the associative memory of a continuous Hopfield network [21, 22] as various attractors to reconstruct the input tokens Ξ^t. In particular, we employ the Hopfield energy function to evolve the input tokens into more globally meaningful representations with respect to these attractors. The attractors $f_{\text{LT}}(\gamma^{t+1})$ are shared across instances of the input tokens $\xi^t \in \Xi^t$. We reconstruct the input tokens using the continuous Hopfield network and the learned attractors in explicit memory as follows:

$$E(\Xi^t) = -\text{lse}(\beta, f_{\text{LT}}(\gamma^{t+1})\Xi^t) + \frac{1}{2}\Xi^t\Xi^{t^T} + \\ \beta^{-1}\log M + \frac{1}{2}(\max_i |f_{\text{LT}}(\gamma_i^{t+1})|)^2, \tag{5}$$

$$\hat{\Xi}^t = \arg\min_{\Xi^t} E(\Xi^t), \tag{6}$$

where t is the batch time step, β is an inverse temperature variable, lse is the log-sum-exp function, and $\max_i |f_{\text{LT}}(\gamma_i^{t+1})|$ represents the largest norm of attractors. We employ an iterative update procedure for a fixed number of steps. All tokens reach their minimum at the same time in the associative memory with their global energy decreasing. For training efficiency, we employed a single step

to reconstruct the tokens as suggested in [22]. Thus, the reconstructed tokens $\hat{\Xi}^t \in \mathbb{R}^{(B \times N) \times E}$ are transformed into $\hat{V}^t \in \mathbb{R}^{B \times N \times E}$ as the output of the global workspace layer. Moreover, depending on β, the reconstructed token $\hat{\Xi}^t$ can be either metastable states representing mixtures of attractors or fixed states represented by one of the attractors. We discuss the tuning of β in Appendix H and demonstrate the complete algorithm in Appendix D.

4. Experiments

In this section, we present the settings and extensive experiment results for classification and relational reasoning tasks. We compare AiT with a broad range of conventional sparse Transformers in terms of model performance and parameter efficiency. Comprehensive ablation studies demonstrate the effectiveness of the bottleneck attention and associative memory for enhanced model performance.

4.1. Settings

Datasets We evaluate model performance based on three different types of datasets: (1) small classification tasks (Triangle [13], CIFAR10, and CIFAR100 [36]), (2) medium and large-sized classification tasks (Oxford Pet [37] and ImageNet100 [38]), and (3) relational reasoning tasks (Sort-of-CLEVR [14]). We train the model on these datasets from scratch using the training split and evaluate using the test split. A detailed description of the datasets can be found in Appendix A.

Model variants We investigate three different sizes of model configurations, i.e., Small, Medium, and Base. The Base variant setting employs 12 layers, 12 attention heads for each layer, a hidden dimension of 768, and an MLP dimension of 3072. The Medium variant has 6 layers, and the Small variant has 2 layers. In what follows, we use brief notation to indicate the model size, for instance, AiT-B means the "Base" variant of the Associative Transformer. Note that the added Global Workspace Layer is light-weight using 0.45M parameters.

Hyperparameters The hyperparameters were chosen based on a grid search. A batch size of 512 was employed for the CIFAR datasets and the Triangle dataset, 128 for the Pet dataset, and 64 for the Sort-of-CLEVR dataset. We utilized the AdamW optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of 0.01. A cosine learning rate scheduler was implemented with an initial learning rate of 1e-5, a warm-up phase of 5 (15) epochs within a total of 100 (300) epochs, and a minimum learning rate set to 1e-6. The smoothing factor of the exponentially weighted moving average, the coefficient σ, and the small value ϵ in the bottleneck balance loss were set to 0.1, 1e-2, and 1e-10, respec-

tively. We employed a bottleneck size of 512 for CIFAR and Pet, 64 for Triangle, and 256 for Relational Reasoning. We utilized 32 memory slots for CIFAR, Triangle, and Relational Reasoning, and 128 slots for Pet. Unless otherwise noted, we trained the model for 100 epochs and reported the mean of three individual experiments. All experiments were based on PyTorch and four A100 GPUs, and the code would be made publicly available. The detailed experimental settings and hyperparameters are presented in Appendix C.

4.2. Classification tasks

The experiments on image classification tasks include comparisons to a broad range of methods. We employed the author-recommended hyperparameters to re-implement these methods. In particular, for the Coordination method [13], we investigated the various model configurations: (1) Coordination consists of 4 layers with parameter sharing among different attention layers, (2) Coordination-D is a deeper model with 8 layers using parameter sharing, (3) Coordination-H is a high memory model with 4 layers that employ individual parameters, and (4) Coordination-DH is a high memory model with 8 layers.

Table 1 demonstrates that AiT performance increased when scaling it from Small to Base, while the Coordination method could not scale with the increasing parameter sizes. AiT achieved better performance compared to Vision Transformers and the other sparse Transformers including Perceiver, Set Transformer, BRIMs, and Luna. Moreover, AiT relies on significantly fewer layers and parameters while outperforming Vision Transformers. Compared to ViT-Base using 12 layers and 85.7M parameters, AiT-Medium has only 6 layers and 45.9M parameters. Nevertheless, AiT-Medium exhibited an average performance of 81.58%, surpassing ViT-Base's performance of 80.46%. Table 2 shows additional comparison between AiT and ViT on the ImageNet100 dataset. AiT achieved superior performance in the various classification tasks using a much smaller parameter size.

Furthermore, compared to ViT-Base with 5.77×10^9 FLOPs during training, AiT-Medium utilized only 2.89×10^9 FLOPs. We compared with the other baseline methods, where the Coordination does not scale well with more model complexity and FLOPs, resulting in degraded performance. In contrast, the proposed Global Workspace Layer is a lightweight architecture, greatly enhancing the classification task performance and parameter efficiency of the conventional sparse Transformers.

We extended the evaluation to the Oxford Pet dataset [37]. We trained the model from scratch for 300 epochs. Figure 2 demonstrates that AiT-Small with only additional 0.9M parameters (two Global Workspace Layers) increased ViT-Small's performance by 13%. Although ViT-Base is a

Methods	CIFAR10 (%)	CIFAR100 (%)	Triangle (%)	Average (%)	Size (M)	#FLOPs
AiT-Base	**85.44 ± 0.31**	**60.78 ± 0.25**	**99.64 ± 0.14**	**81.95**	91.0	5.77×10^9
AiT-Medium	84.59 ± 0.27	60.58 ± 0.32	99.57 ± 0.16	81.58	45.9	2.89×10^9
AiT-Small	83.34 ± 0.44	56.30 ± 0.38	99.47 ± 0.09	79.70	15.8	9.64×10^8
Coordination [13]	75.31 ± 0.72	43.90 ± 0.21	91.66 ± 0.56	70.29	2.2	1.46×10^8
Coordination-DH [13]	72.49 ± 0.61	51.70 ± 0.75	81.78 ± 0.59	68.66	16.6	3.15×10^8
Coordination-D [13]	74.50 ± 0.68	40.69 ± 0.39	86.28 ± 0.73	67.16	2.2	2.91×10^8
Coordination-H [13]	78.51 ± 0.58	48.59 ± 0.72	72.53 ± 0.35	66.54	8.4	1.59×10^8
ViT-Base [2]	83.82 ± 0.17	57.92 ± 0.40	99.63 ± 0.15	80.46	85.7	5.60×10^9
ViT-Medium [2]	82.41 ± 0.11	55.78 ± 0.09	99.62 ± 0.04	79.27	42.7	2.81×10^9
ViT-Small [2]	79.53 ± 0.36	53.19 ± 0.37	99.47 ± 0.07	77.40	14.9	9.36×10^8
Perceiver [26]	82.52 ± 0.82	52.64 ± 0.44	96.78 ± 0.32	77.31	44.9	2.37×10^9
Set Transformer [23]	73.42 ± 0.43	40.19 ± 0.53	60.31 ± 0.29	57.97	2.2	1.11×10^8
BRIMs [39]	60.10 ± 0.50	31.75 ± 0.28	58.34 ± 0.43	50.06	4.4	1.43×10^8
Luna [25]	47.86 ± 0.53	23.38 ± 0.06	57.26 ± 0.19	42.83	77.6	5.08×10^9

Table 1. Performance comparison in the classification tasks.

Methods	Test Accuracy (%)	Size (M)
AiT-Medium	**36.72**	45.9
AiT-Small	33.84	15.8
ViT-Base	34.62	85.7
ViT-Medium	31.72	42.7
ViT-Small	28.16	14.9

Table 2. Performance comparison between AiT and ViT on the ImageNet100 dataset.
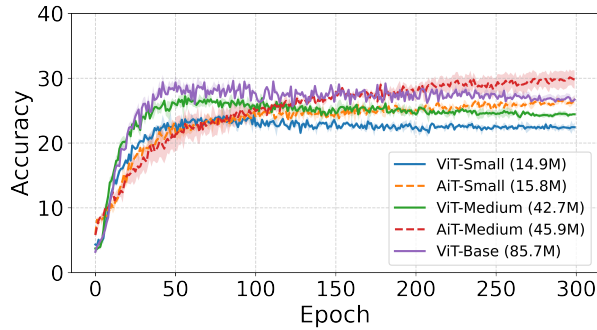

Figure 2. Comparison on the Pet dataset. AiT-Medium demonstrated a stable increase in performance outperforming the ViT-Base.

much larger model with 85.7M parameters with the optimized model regularizations, its accuracy quickly dropped after 50 epochs. In contrast, AiT-Medium with 45.9M parameters demonstrated a stable increase in performance and outperformed the ViT-Base model.

4.3. Ablation study

A comprehensive ablation study is aimed to acquire insights into the functionalities of different components in AiT. In particular, we conducted the following ablations:

- **Reset Memory:** The priors in the explicit memory are initialized every epoch.
- **W/O Hopfield:** The Hopfield network is removed and replaced with cross-attention that shares the same architecture as the multi-head cross-attention in Figure 1.a. The cross-attention takes the input Ξ^t as the query and the upscaled priors $f_{\text{LT}}(\gamma^{t+1})$ as the key and value to compute the output $\hat{\Xi}^t = \text{MHA}(\Xi^t, f_{\text{LT}}(\gamma^{t+1}))$. The rationale behind this ablation is grounded in studies that relied on iterative cross-attentions, such as Set Transformer.
- **W/O Memory:** The global workspace layer is removed.
- **W/O Attention Balance Loss:** The bottleneck is implemented without the attention balance loss.
- **W/O Bottleneck:** The top-k bottleneck attention is replaced with dense pair-wise attention.
- **W/O SA:** The self-attention in Figure 1.b is removed.

Table 3 demonstrates that the bottleneck plays a significant role in enhancing performance, where its absence led to a sizable decrease in accuracy. The 'W/O Attention Balance Loss' ablation presents the effectiveness of the proposed bottleneck attention balance loss. Ablating the other components, such as the Hopfield networks and the explicit memory, while not as impactful, still resulted in degraded accuracy. Consequently, the complete model with all the components achieved the best performance. Additionally, for the 'W/O Hopfield' ablation, Table 4 demonstrates the training time #FLOPs of the Hopfield network compared to the cross-attention component. The Hopfield network is a lightweight component utilizing less than 0.84% FLOPs of the proposed AiT model.

Models	CIFAR10 (%)	CIFAR100 (%)	Triangle (%)	Average (%)
AiT	**83.34 ± 0.44**	**56.30 ± 0.38**	**99.47 ± 0.09**	**79.70**
Reset Memory	81.94 ± 0.41	55.96 ± 0.39	99.46 ± 0.04	79.12
W/O Attention Balance Loss	81.89 ± 0.39	54.72 ± 0.33	99.44 ± 0.08	78.68
W/O Hopfield	81.03 ± 0.45	54.96 ± 0.28	99.44 ± 0.03	78.48
W/O Memory	79.53 ± 0.36	53.19 ± 0.37	99.47 ± 0.07	77.40
W/O Bottleneck	75.40 ± 0.48	46.53 ± 0.41	93.33 ± 0.15	73.75
W/O SA	72.72 ± 0.57	47.75 ± 0.31	99.46 ± 0.04	73.31

Table 3. The ablation study demonstrated that leveraging all the components resulted in the best performance.

Models	#FLOPs
AiT	9.64×10^8
Hopfield network component	8.02×10^6
W/O Hopfield (replaced with cross-attention)	1.19×10^9

Table 4. Hopfield network FLOPs.

Initialization	Accuracy (%)
Gaussian distribution	**83.34 ± 0.44**
Uniform distribution [31]	81.92 ± 0.17
Identity distribution [13]	78.56 ± 0.29

Table 5. Memory initialization methods.

4.4. Explicit memory initialization

In the explicit memory, we initialized priors with values drawn from a specific distribution, including the Gaussian distribution, the uniform distribution [31], and the identity distribution [13]. In particular, the Gaussian distribution generates random values with a mean of zero and a variance of one. The uniform distribution utilizes an upper bound of $\frac{1}{\sqrt{M+D}}$, where M denotes the number of priors, and D is the dimension of a prior. The identity distribution assigns ones on the diagonal and zeros elsewhere. Table 5 indicates that the Gaussian distribution resulted in the best performance, which potentially prevents specific priors from dominating the learning process, for a more balanced memory update in the early training stages.

4.5. Parameter efficiency

Benefits of incorporating the Global Workspace Layer include enhanced model performance and parameter efficiency. We conducted extensive experiments on AiT's parameter efficiency by measuring the model size and test accuracy in CIFAR-10. In particular, we compared a single AiT block to a Coordination block that consists of two multi-head cross-attention components. We explored various configurations by replacing the cross-attention components in the Coordination with the proposed low-rank memory and Hopfield network in AiT.
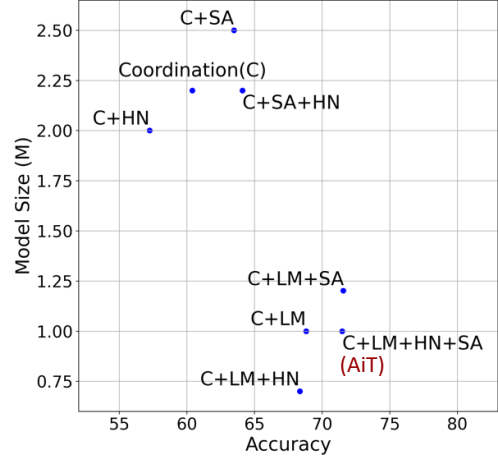

Figure 3. Model size vs. test accuracy for various model configurations. Consolidating all the components in the Coordination block resulted in the best performance of 71.49% maintaining a compact model size of 1.0M.

In Figure 3, 'C' denotes the default Coordination block, 'LM' denotes replacing the first cross-attention in the Coordination block with the low-rank explicit memory, 'HN' denotes replacing the second cross-attention with the Hopfield network, and 'SA' denotes the self-attention component. Compared to the Coordination (C) which achieved an accuracy of 60.41% with a model size of 2.2M, integrating the low-rank memory (C+LM) significantly improved accuracy while reducing the model size to 1M. Moreover, the Hopfield network (HN) achieved competitive performance in C+SA+HN and C+LM+HN+SA configurations for enhanced parameter efficiency. In addition, HN appeared to be effective only when the LM or SA component was employed. This is attributed to the retrieval within the associative memory of the Hopfield network relying on learning a diverse set of priors, which are learned based on the low-rank memory and the self-attention components. Removing these components decreases the model's ability to learn meaningful priors. Consequently, consolidating all the three components in the Coordination block

Methods	Relational (%)	Non-relational (%)
Transformer based models		
AiT-Base	**80.03**	**99.98**
AiT-Medium	78.14	99.75
AiT-Small	76.82	99.85
Coordination [13]	73.43	96.31
ViT-Base	63.35	99.73
ViT-Medium	54.71	99.70
ViT-Small	51.75	98.80
Set Transformer [23]	47.63	57.65
Non-Transformer based models		
CNN+LSTM [14]	60.08	99.47
Dense-Base	46.93	57.71
Dense-Small	47.28	57.68

Table 6. Relational reasoning tasks.

(C+LM+HN+SA), i.e., an AiT block, resulted in the best performance of 71.49% maintaining a compact size of 1.0M. AiT is a sparse representation learner, surpassing the Coordination method in terms of parameter efficiency.

4.6. Prior specialization

The bottleneck attention balance loss encourages competition facilitating the selection of diverse tokens through the bottleneck. We aim to investigate whether the bottleneck attention could contribute to specialized priors that guide attention to various localized features for long-range association. To quantitatively measure the diversity of the selected tokens, we computed the ratio of distinct instances in the selected tokens. Notably, for each prior in the explicit memory, we visualized its attention scores computed by Equation 2 over the different input token positions. Figure 7 presents the corresponding image patches for the selected tokens, with an apparent increase in the patches' diversity as training progresses. Additionally, Figure 8 highlights these attended patches in the image samples where the learned priors demonstrated emergent spatial specialization, dividing labor by focusing on various aspects of the images.

4.7. Relational reasoning tasks

The specialization in the spatial attention allocation of priors results in AiT's enhanced performance in relational reasoning tasks. The aim is to answer questions about the properties and relations of various objects in a given image. We evaluated AiT's performance in the Sort-of-CLEVR dataset [14] for the relational reasoning tasks, which comprises both non-relational and relational tasks. In the non-relational tasks, the question involves the attributes of a specific object, whereas in the relational tasks, the question focuses on the relations between different objects.

To tackle the Sort-of-CLEVR task, we embedded a question using a learnable linear projection, with layer normal-

ization employed before and after the embedding. Then, the question embedding was concatenated to the image patch embeddings as the input to AiT. We compared to both Transformer-based models including Set Transformer and Coordination, and other non-Transformer based models including CNN+LSTM [14] and Dense networks. In CNN+LSTM, images were processed with a CNN to produce a set of objects, and questions were processed with an LSTM to generate a question embedding. Then, an MLP combined the image and question embeddings to predict the answer of the question. Dense-Small and Dense-Base are two fully-connected models derived from AiT-Small and AiT-Base. Table 6 demonstrates AiT's superior performance in the relational and non-relational tasks compared to the Coordination and Set Transformer, achieving a competitive accuracy of 80.03% in the relational tasks. In addition, AiT outperformed the non-Transformer CNN+LSTM model, which relied on a stronger assumption of the input data, i.e., employing the CNN and LSTM for the images and questions, respectively. By contrast, AiT utilized the concatenated tokens of images and questions for the same operations in the proposed Global Workspace Layer.

5. Conclusions

We proposed the Associative Transformer (AiT) based on the bottleneck attention in the explicit memory and the token reconstruction within associative memory to enhance long-range associations of sparsely attended tokens in sparse Transformers. The extensive experiment results demonstrated AiT's enhanced performance and parameter efficiency in the various classification and relational reasoning tasks, surpassing the Vision Transformers and conventional sparse Transformers. We are excited about inducing associative memory for localized contextual learning in sparse Transformers through the set of diverse priors and the continuous Hopfield networks for various tasks.

Limitations Scaling AiT to larger dataset training without fine-tuning the Global Workspace Layer's hyperparameters such as the Hopfield inverse temperature remains a challenge. Implementing a mechanism to adjust adaptively could enhance model performance as it scales. We aim to investigate the countermeasure for more complex tasks, including the CLEVER [40] and bAbI [41] datasets in the future. AiT enables more accessible and cost-effective Transformer model training, showing promise for applications in other domains, such as audio and video. However, potential risks include susceptibility to targeted adversarial attacks, such as Trojan samples exploiting the sparse bottleneck attention mechanism. Exploring potential countermeasures is crucial for its safe deployment in real-world applications.

Acknowledgment

References

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, and et al. Attention is all you need. *NeurIPS*, pages 5998–6008, 2017. 1

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 1, 6, 12

[3] Rodney A. Brooks. Intelligence without representation. *Artif. Intell.*, 47(1-3):139–159, 1991. 1

[4] Klaus Greff, Sjoerd van Steenkiste, and Jürgen Schmidhuber. On the binding problem in artificial neural networks. *arXiv preprint arXiv:2012.05208*, 2020. 1

[5] Bernard J. Baars. *A Cognitive Theory of Consciousness*. Cambridge University Press, 1988. 1

[6] Stanislas Dehaene, Michel Kerszberg, and Jean Changeux. A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Sciences*, 95(24):14529–14534, 1998.

[7] Rufin VanRullen and Ryota Kanai. Deep learning and the global workspace theory. *arXiv preprint arXiv:2012.10390*, 2020.

[8] Arthur Juliani, Kai Arulkumaran, Shuntaro Sasai, and Ryota Kanai. On the link between conscious function and general intelligence in humans and machines. *Transactions on Machine Learning Research*, 2022. 1

[9] Edward Awh, Edward K Vogel, and S-H Oh. Interactions between attention and working memory. *Neuroscience*, 139(1):201–208, 2006. 1

[10] Adam Gazzaley and Anna C Nobre. Top-down modulation: bridging selective attention and working memory. *Trends in cognitive sciences*, 16(2):129–135, 2012. 1

[11] Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000. 1

[12] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *Proceedings of the Royal Society A*, 478(2266):20210068, 2022. 1

[13] Anirudh Goyal, Aniket Rajiv Didolkar, Alex Lamb, and et al. Coordination among neural modules through a shared global workspace. In *ICLR*, 2022. 1, 2, 5, 6, 7, 8, 11, 12

[14] Adam Santoro, David Raposo, David G. T. Barrett, and et al. A simple neural network module for relational reasoning. In *NIPS*, pages 4967–4976, 2017. 1, 5, 8, 11

[15] Murray Shanahan, Kyriacos Nikiforou, Antonia Creswell, Christos Kaplanis, David Barrett, and Marta Garnelo. An explicitly relational neural network architecture. In *International Conference on Machine Learning*, pages 8593–8603. PMLR, 2020.

[16] Sarthak Mittal, Sharath Chandra Raparthy, Irina Rish, Yoshua Bengio, and Guillaume Lajoie. Compositional attention: Disentangling search and retrieval. *arXiv preprint arXiv:2110.09419*, 2021.

[17] Alex F Spies, Alessandra Russo, and Murray Shanahan. Sparse relational reasoning with object-centric representations. *arXiv preprint arXiv:2207.07512*, 2022.

[18] Giancarlo Kerg, Sarthak Mittal, David Rolnick, Yoshua Bengio, Blake Richards, and Guillaume Lajoie. On neural architecture inductive biases for relational tasks. *arXiv preprint arXiv:2206.05056*, 2022.

[19] Jindong Jiang, Fei Deng, Gautam Singh, and Sungjin Ahn. Object-centric slot diffusion. *arXiv preprint arXiv:2303.10834*, 2023.

[20] Shanka Subhra Mondal, Jonathan D Cohen, and Taylor W Webb. Slot abstractors: Toward scalable abstract visual reasoning. *arXiv preprint arXiv:2403.03458*, 2024. 1

[21] John J. Hopfield. Hopfield network. *Scholarpedia*, 2(5):1977, 2007. 1, 2, 4

[22] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, and et al. Hopfield networks is all you need. In *ICLR*, 2021. 1, 2, 4, 5, 12, 13

[23] Juho Lee, Yoonho Lee, Jungtaek Kim, and et al. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, pages 3744–3753, 2019. 2, 6, 8, 12

[24] Ankit Gupta and Jonathan Berant. GMAT: global memory augmentation for transformers. *arXiv preprint arXiv:2006.03274*, 2020. 12

[25] Xuezhe Ma, Xiang Kong, Sinong Wang, and et al. Luna: Linear unified nested attention. In *NeurIPS*, pages 2441–2453, 2021. 2, 6, 12

[26] Andrew Jaegle, Felix Gimeno, Andy Brock, and et al. Perceiver: General perception with iterative attention. In *ICML*, volume 139 of *Proceedings of Machine Learning Research*, pages 4651–4664. PMLR, 2021. 2, 6, 12

[27] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, and et al. Perceiver IO: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 2, 12

[28] Jiezhong Qiu, Hao Ma, Omer Levy, and et al. Block-wise self-attention for long document understanding. In *EMNLP (Findings)*, pages 2555–2565, 2020. 2, 12

[29] Simiao Zuo, Xiaodong Liu, Jian Jiao, and et al. Taming sparsely activated transformer with stochastic experts. In *ICLR*, 2022. 2

[30] James Urquhart Allingham, Florian Wenzel, Zelda E. Mariet, and et al. Sparse moes meet efficient ensembles. *Transactions on Machine Learning Research*, 2022. 2

[31] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv preprint arXiv:1410.5401*, 2014. 2, 7

[32] Dmitry Krotov and John J. Hopfield. Dense associative memory for pattern recognition. In *NIPS*, pages 1172–1180, 2016.

[33] Çaglar Gülçehre, Sarath Chandar, Kyunghyun Cho, and Yoshua Bengio. Dynamic neural turing machine with continuous and discrete addressing schemes. *Neural Comput.*, 30(4), 2018. 2

[34] M Demircigil, J Heusel, M L"owe, S Upgang, and F Vermet. On a model of associative memory with huge storage capacity. *Journal of Statistical Physics*, 168(2):288–299, 2017. 2

[35] Benjamin Hoover, Yuchen Liang, Bao Pham, and et al. Energy transformer. *arXiv preprint arXiv:2302.07253*, 2023. 2

[36] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, 2009. 5, 11

[37] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, pages 3498–3505, 2012. 5, 11

[38] 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. Ieee, 2009. 5

[39] Sarthak Mittal, Alex Lamb, Anirudh Goyal, Vikram Voleti, Murray Shanahan, Guillaume Lajoie, Michael Mozer, and Yoshua Bengio. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *International Conference on Machine Learning*, pages 6972–6986. PMLR, 2020. 6, 12

[40] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, and et al. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, pages 1988–1997, 2017. 8, 11

[41] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M Rush, Bart Van Merriënboer, Armand Joulin, and Tomas Mikolov. Towards ai-complete question answering: A set of prerequisite toy tasks. *arXiv preprint arXiv:1502.05698*, 2015. 8