Associative Transformer Is A Sparse Representation Learner

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

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

Abstract

Emerging from the monolithic pairwise attention mechanism in conventional Transformer models, there is a growing interest in leveraging sparse interactions that align more closely with biological principles. Approaches including the Set Transformer and the Perceiver employ cross-attention consolidated with a latent space that forms an attention bottleneck with limited capacity. Building upon recent neuroscience studies of the Global Workspace Theory and associative memory, we propose the **A**ssoc**i**ative **T**ransformers (AiT). AiT induces low-rank explicit memory that serves as both priors to guide bottleneck attention in shared workspace and attractors within associative memory of a Hopfield network. We show that AiT is a sparse representation learner, learning distinct priors through the bottlenecks that are complexity-invariant to input quantities and dimensions. AiT demonstrates its superiority over methods such as the Set Transformer, Vision Transformer, and Coordination in various vision tasks.

1 Introduction

The predominant paradigm in conventional deep neural networks has been characterized by a monolithic architecture, wherein each input sample is subjected to uniform processing within a singular model framework. For instance, Transformer models use pairwise attention to establish correlations among disparate segments of input information [1, 2]. Emerging from the pair-wise attention mechanism, there is a growing interest in leveraging modular and sparse interactions that have been shown to align more closely with biological principles [3, 4, 5, 6].

Modularization of knowledge can find resonance with the neuroscientific grounding of the Global Workspace Theory (GWT) [7, 8, 9, 10]. GWT explains a fundamental cognitive architecture for information processing within the brain. The bottleneck facilitates the processing of content-addressable information through attention that is guided by working memory [11, 12]. The coordination method [13] represents the initial attempt to assess the effectiveness of GWT in conventional neural network models. Unfortunately, this method relies on iterative cross-attention for both information writing and retrieval. In the human brain, it is evident that memory typically encompasses both working memory and long-term memory. Specifically, the hippocampus operates on Hebbian learning for retrieving information from working memory, akin to the associative memory found in Hopfield networks [14, 15]. Our research has revealed that replacing such a repetitive attention-based mechanism with a consolidated, more biologically-plausible associative memory can lead to improved performance. Our objective is to introduce a shared workspace augmented with associative memory into Transformers, thereby facilitating a more comprehensive and efficient association of information fragments.

To this end, we propose the **A**ssoc**i**ative **T**ransformer (AiT) based on a novel global workspace layer augmented by associative memory. The global workspace layer entails three main components: 1) the squash layer: input data is transformed into a list of patches regardless of which samples they come

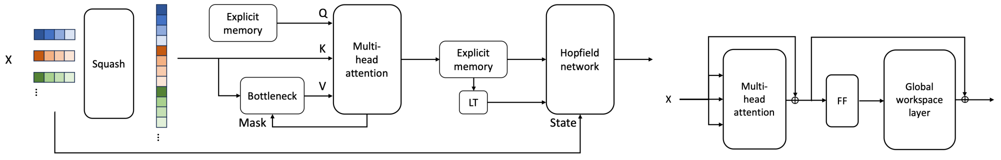
[*] Both authors contributed equally. Corresponding author: ywsun@g.ecc.u-tokyo.ac.jp

from, 2) the bottleneck attention: patches are sparsely selected to learn a set of priors in low-rank memory based on a bottleneck attention mechanism, and 3) the Hopfield network: information is broadcast from the shared workspace to update the current input based on associative memory of a Hopfield network. Through end-to-end training, we show the emerging specialization of priors, contributing to enhanced performance in vision tasks over conventional approaches including Set Transformer, Vision Transformer, Perceiver, and Coordination.

2 Associative Transformer

2.1 Global workspace layer

We devise an associative memory-augmented attention layer called the *global workspace layer*, which comprises the squash layer, the bottleneck attention guided by low-rank explicit memory, and the information retrieval within the associative memory of a Hopfield network (Figure 1). We refer to Appendix A.2 for the essential definitions of the attention mechanism and Vision Transformers.



(a) Global workspace layer

(b) Associative Transformer block

Figure 1: The scheme of the Associative Transformer.

Squash layer In pairwise self-attention, patches from the same sample are attended to. We improve the diversity in patch-wise correlation learning beyond one sample using a *squash layer*. The squash layer concatenates N patches $V \in \mathbb{R}^{B \times N \times E}$ within one batch B into vectors $V \in \mathbb{R}^{(B \times N) \times E}$, which forms a list of patches regardless of the samples they are from. Since the bottleneck with a fixed capacity k decreases the complexity from $O((B \times N)^2)$ to $O((B \times N) \times k)$, using the squash layer will not add up to the complexity, but increase the diversity of input patches.

Low-rank explicit memory An explicit memory bank with limited slots aims to learn M *priors* $\gamma = \mathbb{R}^{M \times D}$ where D is the dimension of the prior. The priors in the memory bank are used as various keys to compute the bottleneck attentions that extract different sets of patches from the squashed input. Further using low-rank priors can reduce memory consumption, whose lower dimension $D << E$ is obtained by applying a down-scale linear transformation on the input patch dimension E.

2.2 Bottleneck attention with a limited capacity

The objective of the bottleneck attention is to learn a set of priors that guide attention to various sets of input patches. This is enabled by a cross-attention mechanism constrained by hard attention. We first consider a tailored cross-attention mechanism to update the memory bank based on the squashed input $\Xi^t = V^t \in \mathbb{R}^{(B \times N) \times E}$, then we discuss the case of limiting the capacity via a top-k hard attention. Notably, in the cross-attention, the query is a function W^Q of the current memory content $\gamma^t = \{\gamma_i^t\}_{i=1}^M$. The key and value are functions W^K and W^V of the squashed input Ξ^t. The attention scores for head i can be computed by $\mathrm{A}_i^t(\gamma^t, \Xi^t) = \mathrm{softmax}(\frac{\gamma^t W_{i,t}^Q (\Xi^t W_{i,t}^K)^T}{\sqrt{D}})$.

The hard attention allows patches to compete to enter the workspace through a k-size bottleneck, fostering essential patches to be selected. In particular, we select the top-k patches with the highest attention scores from A_i^t. The priors in the memory are then updated based on the top-k patches. Moreover, to ensure a stable update of these priors across different time steps, we employ the layer normalization and the exponentially weighted moving average as follows

$$\mathrm{head}_i^t = \text{top-}k(\mathrm{A}_i^t)\Xi^t W_t^V, \ \hat{\gamma}^t = \mathrm{LN}(\mathrm{Concat}(\mathrm{head}_1^t, \ldots, \mathrm{head}_A^t)W^O), \tag{1}$$

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

where top-k is a function to select the k highest attention scores and set the other scores to zero, LN is the layer normalization function, and α is a smoothing factor.

Bottleneck attention balance loss Employing multiple global workspace layers cascaded in depth leads to the difficulty in the emergence of specialized priors in the explicit memory (Figure 6). To overcome this challenge, we propose the bottleneck attention balance loss to encourage the selection of diverse patches from different input positions in the shared workspace. $\ell_{\text{bottleneck}}$ comprises two components, i.e., the accumulative attention scores and the chosen instances for each input position. Then, we derive the normalized variances of the two metrics across different input positions as follows

$$\ell_{\text{loads}_{i,l}} = \sum_{j=1}^{M}(\mathrm{A}_{i,j,l}^{t} > 0),\ \ell_{\text{importance}_{i,l}} = \sum_{j=1}^{M}\mathrm{A}_{i,j,l}^{t}, \tag{3}$$

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

where $\mathrm{A}_{i,j,l}^{t}$ denotes the attention score of the input position l for the jth memory slot of head i, $\ell_{\text{importance}}$ represents the accumulative attention scores in all M memory slots for each input position, ℓ_{loads} represents the accumulative chosen instances for each input position, $\mathrm{Var}(\cdot)$ denotes the variance function, and ϵ is a small value. The losses for all the heads are summed up as follows $\ell_{\text{bottleneck}} = \sigma \cdot \sum_{i=1}^{A}\ell_{\text{bottleneck}_i}$ where σ is a coefficient.

2.3 Information retrieval within associative memory

Attractors Priors learned in the memory bank act as attractors in associative memory. Attractors have basins of attraction defined by an energy function. Any input state that enters an attractor's basin of attraction will converge to that attractor. The attractors in associative memory usually have the same dimension as input states, however, the priors γ^{t+1} in the memory bank have a lower rank compared to the input. Therefore, we employ a learnable linear transformation $f_{\text{LT}}(\cdot)$ to project the priors into the same dimensional space E of the input before using them as attractors.

Retrieval using an energy function in Hopfield networks Hopfield networks have demonstrated their potential as a promising approach to constructing associative memory. In particular, we use a continuous Hopfield network [16, 15] operating with continuous input and output values. The upscaled priors $f_{\text{LT}}(\gamma^{t+1})$ are stored within the continuous Hopfield network and subsequently retrieved to reconstruct the input state $\Xi^t \rightarrow \hat{\Xi}^t$. Then, we update each patch representation $\xi^t \in \Xi^t$ by decreasing its energy $E(\xi^t)$ within the associative memory as follows

$$E(\xi^t) = -\mathrm{lse}(\beta, f_{\text{LT}}(\gamma^{t+1})\xi^t) + \frac{1}{2}\xi^t{\xi^t}^T + \beta^{-1}\mathrm{log}M + \frac{1}{2}\zeta^2, \tag{5}$$

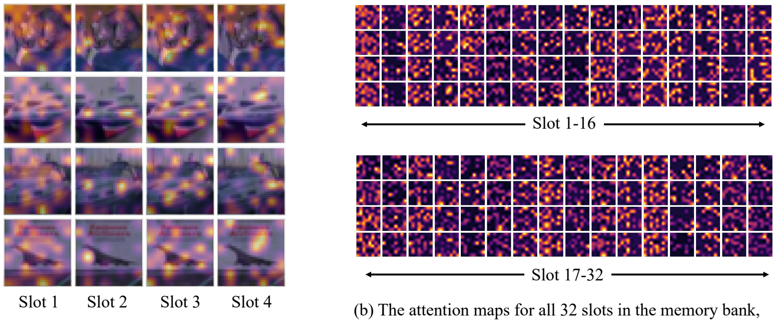
$$\zeta = \max_i |f_{\text{LT}}(\gamma_i^{t+1})|,\ \hat{\xi}^t = \arg\min_{\xi^t} E(\xi^t), \tag{6}$$

where lse is the log-sum-exp function, β is an inverse temperature variable, and ζ denotes the largest norm of attractors. A skip connection functioning as the information broadcast is employed to obtain the final output $\Xi^{t+1} = \hat{\Xi}^t + \Xi^t$.

3 Experiments

Our study demonstrates that AiT outperforms the previously employed attention-based approaches such as the Coordination, the Set Transformer, and the Perceiver when applied to vision-related tasks. The detailed description of the experiment settings can be found in Appendix A.4.

Classification tasks We conducted experiments comparing to a wide range of methods (Table 1). The results show that AiT achieved better performance compared to the coordination methods. Moreover, compared to ViT-Small, AiT-Small exhibited only a marginal increase in the parameter size (0.9M), while improving the performance by 3.81%. AiT also outperformed other sparse attention-based methods of Perceiver and Set Transformer. Figure 3 depicts the results for CIFAR-10 based on models with a single layer. The low-rank memory (LM) showed benefits in both improving the performance and decreasing the model size. The Hopfield network (HN) maintained the performance while reducing the size by replacing the cross-attention with more efficient retrieval. Integrating all three components (C+LM+HN+SA) resulted in a competitive accuracy of 71.49% with a compact size of only 1.0M. Additionally, We extended the evaluation to the Pet dataset in Figure 4.



(a) A glimpse of the attention maps at Slot 1 to Slot 4 of four distinct input images.

(b) The attention maps for all 32 slots in the memory bank, applied to four distinct input images. Each memory slot learned to attend to different regions of pixels in input images.

Figure 2: Learned distinct memory slot attentions in AiT.

Table 1: Performance comparison in image classification tasks

Methods	CIFAR10	CIFAR100	Triangle	Average	Model Size
AiT-Base	**85.44**	**59.10**	99.59	**81.38**	91.0
AiT-Small	83.34	56.30	99.47	79.70	15.8
Coordination [13]	75.31	43.90	91.66	70.29	2.2
Coordination-DH	72.49	51.70	81.78	68.66	16.6
Coordination-D	74.50	40.69	86.28	67.16	2.2
Coordination-H	78.51	48.59	72.53	66.54	8.4
ViT-Base [2]	83.82	57.92	**99.63**	80.46	85.7
ViT-Small	79.53	53.19	99.47	77.40	14.9
Perceiver [17]	82.52	52.64	96.78	77.31	44.9
Set Transformer [18]	73.42	40.19	60.31	57.97	2.2
BRIMs [19]	60.10	31.75	-	45.93	4.4
Luna [20]	47.86	23.38	-	35.62	77.6

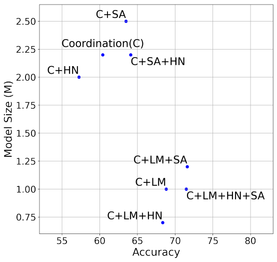


Figure 3: Model size vs. accuracy for configurations.

Ablation study We conducted an ablation study to gain insights into the functionalities of the various components of AiT (Table 2). The detailed settings are in Appendix A.4. The analysis suggested that the complete model with all components enabled achieved the highest accuracy in all the tasks. Notably, the bottleneck component appeared to play a significant role, since its absence led to an evident decrease in test accuracy. Making changes to other components such as Hopfield networks and the explicit memory, while not as impactful, still resulted in degraded accuracy.

Table 2: Comparison based on an ablation study.

Models	CIFAR10	CIFAR100	Triangle	Average
AiT	**83.34**	**56.30**	**99.47**	**79.70**
Reset memory	81.94	55.96	99.46	79.12
W/O Hopfield	81.03	54.96	99.44	78.48
W/O memory (ViT)	79.53	53.19	99.47	77.40
W/O bottleneck	75.40	46.53	93.33	73.75
W/O SA	72.72	47.75	99.46	73.31

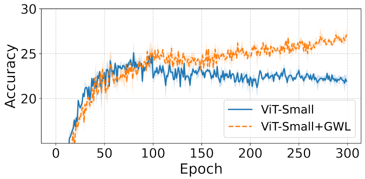


Figure 4: Comparison on the Pet dataset, which shows enhanced accuracy for the proposed method.

Prior specialization Patches in one image can be attended sparsely by different priors through the bottleneck attention. These priors learn to focus on independent spatial areas of an image, facilitating the natural emergence of specialized priors that guide the attention. We visualized the activation maps for the specialized priors in the CIFAR-10 task for the AiT-Small model. In Figure 2, each slot's activation maps are highlighted over a specific area of images during the selection of relevant patches.

Efficacy of Bottleneck Attention Balance Loss The Bottleneck Attention Balance Loss facilitates a diverse selection of the input patches for each prior. To quantitatively measure the efficacy, we computed sparsity scores that represent the ratio of distinct patches in all selected patches. In Figure 7, we can observe an apparent increase in the selected patch diversity.

4 Conclusions

We proposed the Associative Transformer (AiT), an architecture inspired by the Global Workspace Theory from neuroscience studies. AiT integrates a diverse set of priors with the emerging specialization property and associative memory to enhance learning among image patches. The comprehensive experiments demonstrate AiT's efficacy compared to the conventional methods such as the Coordination method. In the future, we aim to investigate multi-modal competition within the shared workspace, enabling tasks to benefit from the cross-modal learning of distinct perception inputs.

References

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

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, and et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

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

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

[5] Danijar Hafner, Alexander Irpan, James Davidson, and Nicolas Heess. Learning hierarchical information flow with recurrent neural modules. In *NIPS*, pages 6724–6733, 2017.

[6] Marvin Minsky. *The Society of Mind*. Simon & Schuster, Inc., 1986.

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

[8] Changeux J. P. Dehaene S., Kerszberg M. A neuronal model of a global workspace in effortful cognitive tasks. In *National Academy of Sciences*, 1998.

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

[10] 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.

[11] E. Awh, E.K. Vogel, and S.-H. Oh. Interactions between attention and working memory. *Neuroscience*, 2006.

[12] Adam Gazzaley and Anna C. Nobre. Top-down modulation: bridging selective attention and working memory. *Trends in Cognitive Sciences*, 2011.

[13] Anirudh Goyal, Aniket Rajiv Didolkar, Alex Lamb, and et al. Coordination among neural modules through a shared global workspace. In *ICLR*, 2022.

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

[15] Hubert Ramsauer, Bernhard Schäfl, Johannes Lehner, and et al. Hopfield networks is all you need. In *ICLR*, 2021.

[16] 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.

[17] Andrew Jaegle, Felix Gimeno, Andy Brock, and et al. Perceiver: General perception with iterative attention. In *ICML*, 2021.

[18] Juho Lee, Yoonho Lee, Jungtaek Kim, and et al. Set transformer: A framework for attention-based permutation-invariant neural networks. In *ICML*, 2019.

[19] Sarthak Mittal, Alex Lamb, Anirudh Goyal, and et al. Learning to combine top-down and bottom-up signals in recurrent neural networks with attention over modules. In *ICML*, 2020.

[20] Xuezhe Ma, Xiang Kong, Sinong Wang, and et al. Luna: Linear unified nested attention. In *NeurIPS*, 2021.

[21] Jiezhong Qiu, Hao Ma, Omer Levy, and et al. Blockwise self-attention for long document understanding. In *EMNLP*, 2020.

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

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

[24] Anirudh Goyal and Yoshua Bengio. Inductive biases for deep learning of higher-level cognition. *arXiv:2011.15091*, 2020.

[25] Dianbo Liu, Alex Lamb, Kenji Kawaguchi, and et al. Discrete-valued neural communication. In *NeurIPS*, 2021.

[26] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, and et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *ICLR*, 2017.

[27] Carlos Riquelme, Joan Puigcerver, Basil Mustafa, and et al. Scaling vision with sparse mixture of experts. In *NeurIPS*, 2021.

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

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

[30] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *arXiv:1410.5401*, 2014.

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

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

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

[34] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

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

[36] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012.

[37] Wenhai Wang, Enze Xie, Xiang Li, and et al. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021.

[38] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, and et al. Levit: a vision transformer in convnet's clothing for faster inference. In *ICCV*, 2021.

A Appendix

A.1 Related work

This section provides a summary of relevant research concerning sparse attention architectures. We investigate and compare these studies based on their relatedness to the global workspace in terms of several key conditions (Table 3). Firstly, we examine whether an architecture involves operations of information writing and reading through shared workspace. Secondly, we assess whether the latent representations (priors) in workspace memory are subsequently processed by self-attention. Thirdly, we inspect whether the latent representations have a lower rank compared to the input representations. Fourthly, we analyze whether the information retrieval from the workspace is driven by a bottom-up or a top-down signal. Lastly, we investigate whether the model incorporates a bottleneck with a limited capacity to regulate the information flow passing through the workspace.

Table 3: Comparison of attention architectures based on properties of the Global Workspace Theory

Method	Operations		Self-Attention	Low-Rank Memory	Top-Down/Bottom-Up	Bottleneck
	Writing	Reading				
Vision Transformer [2]	-	-	-	-	BU	×
BlockBERT [21]	-	-	-	-	BU	✓
BRIMs[19]	×	×	×	×	TD	✓
Modern Hopfield [15]	×	✓	×	✓	BU	×
Perceiver [17]	✓	×	✓	✓	BU	×
Coordination [13]	✓	✓	×	×	BU	✓
Perceiver IO [22]	✓	✓	✓	✓	TD	×
Set Transformer [18]	✓	✓	×	×	BU	×
Luna [20]	✓	✓	×	×	BU	×
GMAT [23]	✓	✓	✓	×	BU	✓
Associative Transformer (Ours)	✓	✓	×	✓	BU	✓

Unlike convolutional neural networks, Transformer models do not possess inductive biases that allow the model to attend to different segments of the input data [24]. To enhance Transformer models, studies of sparse attention architectures explored consolidating latent memory to adaptively extract representations from input data. Utilizing priors such as latent memory in the attention mechanism has demonstrated the advantages of improved model generality [23, 17, 13, 22, 18, 20]. For instance, Perceiver [17] used iterative cross-attention with a latent array as priors and a latent transformation applied to the priors, to capture dependencies across input data. Perceiver IO [22] further incorporated top-down attention to read information from the priors. Set Transformer [18] and Linear Unified Nested Attention (Luna) [20] also employed iterative cross-attention, but without using a latent transformation. Other attention mechanisms that do not involve priors are omitted since these studies usually rely on strong inductive biases with predefined network modularization. For example, Blockwise Self-Attention [21] introduced block masking matrices determined by a set of predefined permutations, for the specialization of different attention heads. In our method, however, distinct priors naturally emerge through end-to-end training. Moreover, the previous methods using latent memory necessitated priors with the same dimension as the input. In contrast, we devise low-rank priors that can be encoded and decoded adaptively based on scaling transformations, which increase the memory capacity. Additionally, the iterative cross-attention used in the previous studies can be replaced with single attention augmented with associative memory in the proposed method.

In the same vein of building sparse attention mechanisms through shared workspace, Coordination [13] used iterative cross-attentions via a bottleneck to encourage more effective module communication, which the previous studies usually do not consider. They argued that more flexibility and generalization could emerge through the competition of specialized modules. A following study [25] of the Coordination method assessed using discrete-valued priors for learning better-refined representations and module specialization. However, the priors in the Coordination methods possessed the same dimension to the input and the number of priors is limited to less than 10. The evaluation was also restricted to simple tasks. Unlike the previous work in Coordination, we propose low-rank explicit memory to learn a larger set of specialized priors (up to 128) from a pool of patches (up to 32.8k). Moreover, the Coordination methods relied on iterative cross-attentions to learn such priors, while this work focuses on a novel learning method of associative memory-augmented attention.

This work is also related to modular neural networks in terms of competition in the shared workspace. Separating the information processing within Transformers into distinct components, depending on the input data, has shown advantages in adding more flexibility in data processing [26, 27, 28, 29]. Unlike the work in modular neural networks, we aim to comprehend the emerging behavior of prior specialization and the interplay among these priors for an enhanced attention mechanism.

Furthermore, external memory such as tape storage and associative memory has been successfully employed in various machine learning tasks [30, 31, 32, 33]. Recent studies explored the potential use of Hopfield networks [14] and their modern variants [16, 15] in relation to Transformers. In contrast to these investigations, we incorporate Hopfield networks as an integral element in constructing the global workspace layer, functioning as a mechanism for the information broadcast in the shared workspace. The goal is fundamentally different from prior research, which focused on using Hopfield networks independently of the attention mechanism.

A.2 Vision Transformers

Vision Transformers (ViT) tackle image classification tasks by processing sequences of image patches. The pre-processing layer partitions an image into a sequence of non-overlapping patches, followed by a learnable linear projection. Let $x \in \mathbb{R}^{H\times W\times C}$ be an image input, 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 then mapped to embeddings $v_p \in \mathbb{R}^{N\times E}$ with the linear projection. ViT leverages self-attention where each head maps 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 is a weighted sum of the values formulated as follows

$$\mathrm{h}^i(v) = \mathrm{softmax}(\frac{W_i^Q v(W_i^K v)^T}{\sqrt{D}})\, W_i^V v, \tag{7}$$

$$\text{Multi-head}(v) = \mathrm{Concat}(\mathrm{h}^1, \ldots, \mathrm{h}^A)\, W^O, \tag{8}$$

where W^O is a linear transformation for outputs, and A is the number of attention heads.

A.3 Inspecting Attention Heads in Vision Transformers

We assume that the competition existing within the pair-wise attention of different patches would be of importance for the model to learn meaningful representations. If such competition exists, a trained model will naturally result in sparser interactions in attention heads. To understand the operating modes of attention heads in a vision Transformer (ViT) model, we measured the interaction sparsity of the pairwise self-attention. The goal is to investigate whether sparse attention is required to learn meaningful representations during the model training. If the ViT model inherently learns such sparse interactions among patches, we can induce an inductive bias to foster the sparse selection of patches through a communication bottleneck. We trained a ViT-Base variant model for 100 epochs from scratch for the CIFAR-10 task. Then, for each attention head, we obtained a violin plot to represent the distribution of attention sparsity for different patches. The attention sparsity for a specific patch's interactions with other patches is computed as follows

$$\arg\min_s \sum_{j=1}^{s} A^{i,j} \geq 0.9, \tag{9}$$

where $A^{i,j}$ is the attention score allocated to the jth patch by the ith patch. The attention sparsity score is measured by the minimal number of required patches whose attention scores add up to 0.90. For instance, there are 65 patches for the CIFAR-10 task with patch size 4, thus 65 interactions for each patch to all the patches including itself. Then, an attention head has a higher sparsity if the median of the required patches s that satisfies Equation 9 across different patches is smaller. Moreover, for each head, we show the distribution of the attention sparsity scores for different patches in the violin plots (Figure 5), where the scores from different input samples were averaged. The number in the center of each panel gives the median $\bar{s}$ of the distribution. The heads in each layer are sorted according to $\bar{s}$. Note that training the model for a longer duration can result in even better convergence and higher attention sparsity. We also refer to a concurrent investigation on the attention sparsity of the Bidirectional Encoder Representations from Transformers (BERT) model for natural language processing (NLP) tasks [15]. Our findings about the various operating modes of attention heads in ViT are in line with the findings in BERT for NLP tasks.

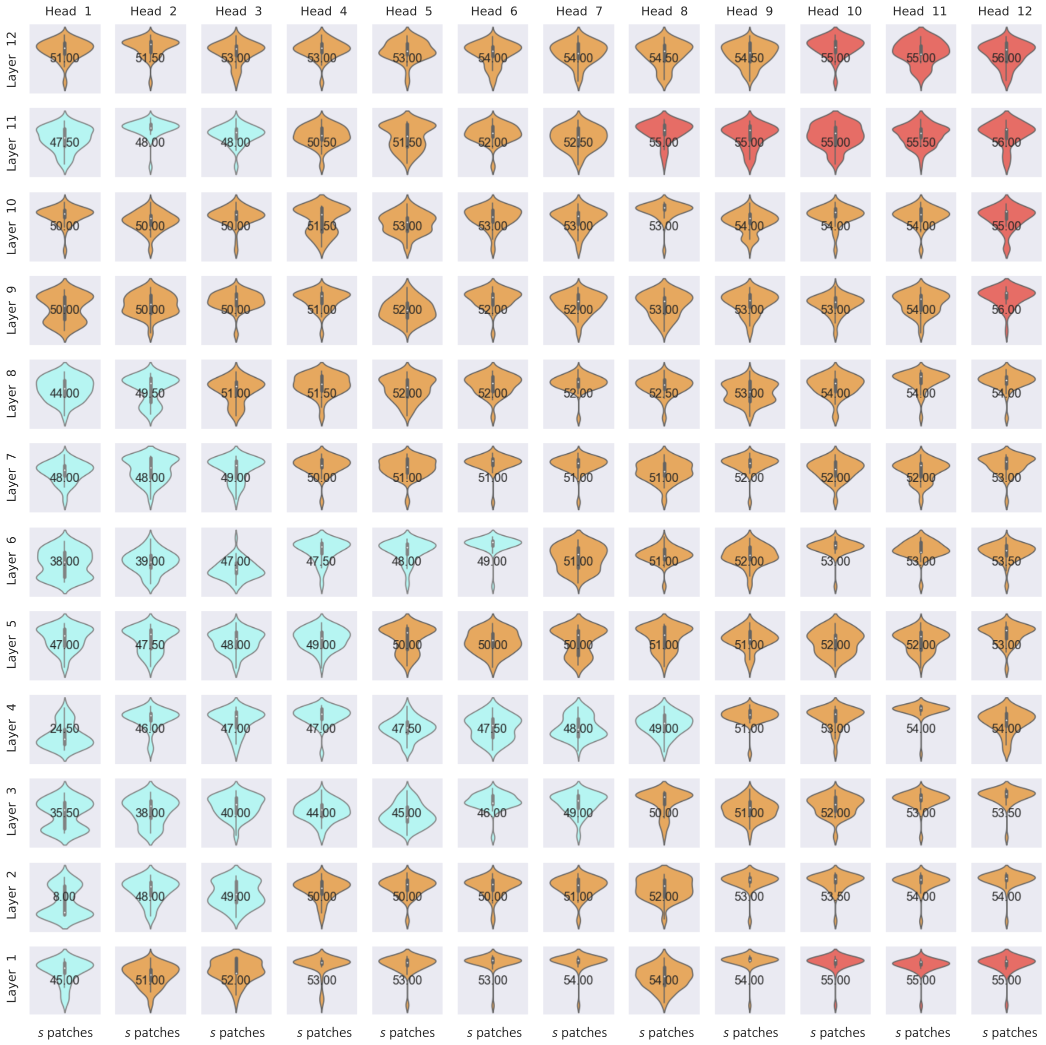


Figure 5: Analysis of operating modes of attention heads in the ViT-Base model. We recognize three different groups of attention heads based on their sparsity scores. Group (I) in light blue: High sparsity heads abundant in the middle layers 3-6. The vast majority of these heads only used 50% or fewer interactions. Group (II) in orange: Middle sparsity heads predominant in layers 2 and 7-10. Less than 80% of the interactions were activated. Group (III) in red: Low sparsity heads observed in high layers 11-12 and the first layer, where the most patches were attended to. Inducing the global workspace layer will provide the inductive bias to attend to the essential patches more effectively.

The inspection of the attention heads revealed the existing competition among patches and a large redundancy in the pair-wise interactions. Notably, less than 80% interactions were activated across different layers and several heads from the middle layers only used 50% or less interactions. Additionally, the middle layers exhibited higher sparsity, compared to the other layers. This is possibly because the middle layers learn more general representations that are usually attended to and leveraged for various tasks. Based on our observation, introducing a communication bottleneck with limited capacity can foster competition among patch representations. This bottleneck imposes constraints on the number of patches that each attention head can focus on, thereby enhancing the inductive bias for meaningful patch learning through this competitive process.

A.4 Experimental settings and hyperparameters

Datasets We describe the applied datasets in this section. (1) CIFAR-10 [34] is an image collection of 10 objects, covering 50k training samples and 10k test samples, labeled as airplane, automobile, and so on. The size of images is $32 \times 32 \times 3$. (2) CIFAR-100 [35] contains 100 object classes with 500 training images and 100 testing images per class. For both CIFAR-10 and CIFAR-100 datasets, we performed the random cropping with size $32 \times 32 \times 3$ and a padding size of 4. (3) Triangle dataset [13] includes 50k training images and 10k test images with size 64×64, each of which contains 3 randomly placed clusters of points. The task is to predict whether the three clusters form an equilateral triangle or not. (4) Oxford-IIIT Pet dataset [36] comprises 37 categories featuring diverse breeds of cats and dogs, with 200 images allocated for each class. We utilized the random resized cropping with size $256 \times 256 \times 3$ and resized all images to size $224 \times 224 \times 3$. Additionally, we applied the random horizontal flip and normalization to the CIFAR-10, CIFAR-100, and Pet datasets.

Model variants We investigate two different sizes of model configurations, i.e., Small and Base. The Base variant setting is adapted from Vision Transformer (ViT) using 12 layers, 12 attention heads for each layer, a hidden dimension of 768, and an MLP dimension of 3072. The Small variant using 2 layers is added for efficient comparison among approaches. The CLS token is removed while the pooled representations of the last dense network are used instead since using the CLS token leads to undermined learning results in the vision tasks [37, 38].

Regarding the coordination method, we have examined the efficacy of its variants with different model configurations. The default coordination model consists of 4 layers, with parameter sharing among attention layers. Coordination-D is a deeper model with 8 layers using parameter sharing. Coordination-H is a high-capacity model with 4 layers that employ individual parameters. Coordination-DH is a high-capacity model with 8 layers.

Hyperparameters Table 4 presents the hyperparameters used for the different tasks in this study. 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 is set to 0.9. The coefficient σ and the small value ϵ in the bottleneck balance loss are set to 1e-2 and 1e-10 respectively.

By default, we employed 8 attention heads and 32 memory slots for the bottleneck attention. To obtain the bottleneck size, we considered two main factors of the batch size and the patch size. For the CIFAR and Pet datasets, we used a bottleneck size of 512, which selected from a pool of 32.8k/25.1k patches. For the Triangle dataset, we used a bottleneck size of 64 from a pool of 2.0k patches. In relation to the bottleneck size and the patch pool size, we used 128 memory slots for the Pet dataset and 32 memory slots for the other datasets. Moreover, we trained the models on the Pet dataset for 300 epochs and on the other datasets for 100 epochs.

Table 4: Hyperparameters

Parameter	**Value**
Common parameters	
Optimizer	AdamW
Weight decay	0.01
Learning rate	1×10^{-4}
Number of self-attention heads	12
Number of attention layers	2 (Small)/ 12 (Base)
Size of hidden layer	768
Size of MLP	3072
Size of memory slot	32
Number of bottleneck attention heads	8
Beta	1.0
Epochs	100 (300 for Oxford Pet)
CIFAR	
Patch size	4
Batch size	512
Number of memory slots	32
Bottleneck size	512
Triangle	
Patch size	32
Batch size	512
Number of memory slots	32
Bottleneck size	64
Oxford Pet	
Patch size	16
Batch size	128
Number of memory slots	128
Bottleneck size	512

Ablation study In AiT with reset memory, we assessed AiT's performance by initializing its explicit memory every epoch. W/O Hopfield evaluated the model performance when we replaced the Hopfield network with another cross-attention mechanism. W/O memory evaluates performance when the global workspace layer is removed, the remaining components of which are equivalent to a simple Vision Transformer. W/O bottleneck estimates performance using dense attention by removing the top-k bottleneck capacity constraint. W/O SA examines performance when the self-attention component is excluded.

A.5 Efficacy of the Bottleneck Attention Balance Loss

The Bottleneck Attention Balance Loss facilitates the learning of priors that can attend to diverse sets of patches. We demonstrate the efficacy by visualizing the bottleneck attention scores computed using the learned priors (Figure 6) and the corresponding selected patches (Figure 7). We used as a metric the ratio of distinct patches in all the selected patches by the bottleneck attention. With the progress of training, we can obtain a more diverse selection of patches.

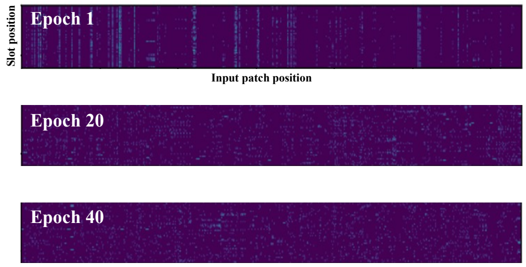


Figure 6: Bottleneck attention balance loss facilitates the selection of diverse patches from different input positions.

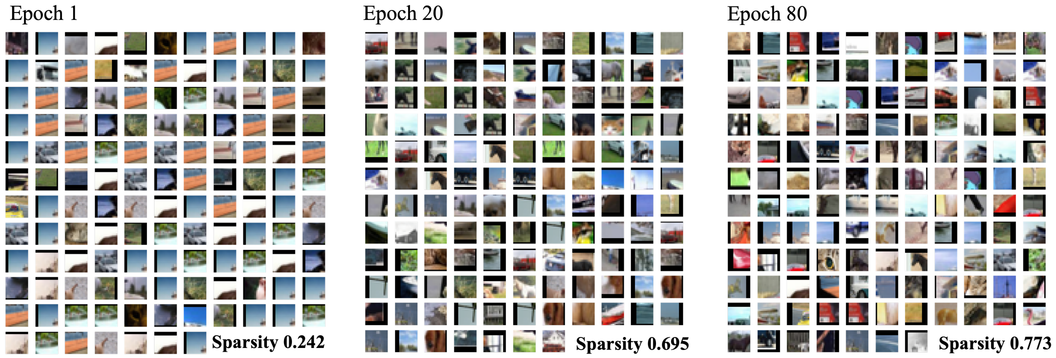


Figure 7: Examples of the selected patches by the bottleneck attention in CIFAR-10. With the progress of training, we can obtain a more diverse selection of patches.

A.6 Hopfield networks energy

The information retrieval in the global workspace layer is based on a continuous Hopfield network, where an input state converges to a fixed attractor point within the associative memory of the Hopfield network. Usually, any input state that enters an attractor's basin of attraction will converge to that attractor. The convergence results in a decreased state energy with respect to the stored attractors in the memory. To quantitatively measure the amount of energy reduction during the information retrieval process in the Hopfield network, we computed an input state's energy before and after it was reconstructed based on the energy function in Equation 5. A successful retrieval will result in a substantial reduction in the state energy during the process (Figure 8).

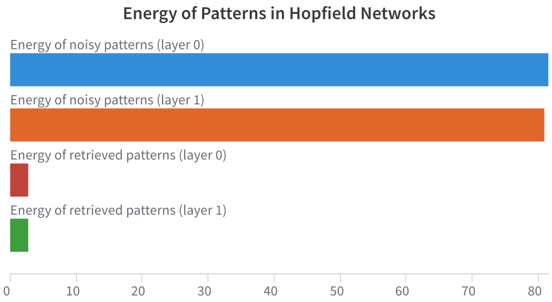


Figure 8: Energy of patch representations in the CIFAR-10 task for AiT-Small. The Hopfield network operates by iteratively decreasing the energy of an input state in relation to the attractors stored in its memory. This reduction in energy enables the retrieval of a representation that closely aligns with previously learned attractors, effectively leveraging knowledge within the associative memory.