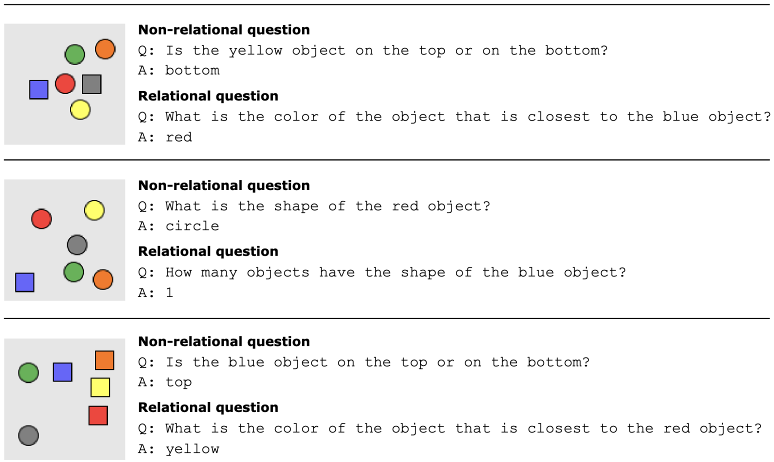
A. Datasets

In this section, we describe the datasets used in this work. (1) CIFAR-10 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 × 32 × 3. (2) CIFAR-100 [36] contains 100 object classes with 500 training images and 100 testing images per class. For both the CIFAR-10 and CIFAR-100 datasets, we performed random cropping with size 32 × 32 × 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 [37] comprises 37 categories featuring diverse breeds of cats and dogs, with 200 images allocated for each class. We utilized random resized cropping with size 256 × 256 × 3 and resized all images to size 224 × 224 × 3. Additionally, we applied random horizontal flip and normalization to the CIFAR-10, CIFAR-100, and Oxford-IIIT Pet datasets. (5) Sort-of-CLEVR dataset [14] is a simplified version of the CLEVR dataset [40]. It includes 10k images with size 75 × 75 × 3 and 20 different questions (10 relational and 10 non-relational questions) for each image. In each image, objects with randomly chosen shapes (square or circle) and randomly chosen colors (red, green, blue, orange, gray, yellow) are placed (Figure 4).



Non-relational question
Q: Is the yellow object on the top or on the bottom?
A: bottom
Relational question
Q: What is the color of the object that is closest to the blue object?
A: red

Non-relational question
Q: What is the shape of the red object?
A: circle
Relational question
Q: How many objects have the shape of the blue object?
A: 1

Non-relational question
Q: Is the blue object on the top or on the bottom?
A: top
Relational question
Q: What is the color of the object that is closest to the red object?
A: yellow

Figure 4. Examples from the Sort-of-CLEVR dataset [14].

B. Comparison of work related to Global Workspace Theory

We discuss and summarize the existing sparse attention methods in relation to the properties of Global Workspace Theory (Table 7). First, we examine whether an architecture involves operations of information writing and reading through a 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 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.

C. Experimental settings and hyperparameters

Table 8 presents the hyperparameters used for the different tasks in this study. Unless otherwise noted, we employed the author-recommended settings and hyperparameters for the re-implementation of baseline models. The small variants of ViT and AiT have 2 attention layers, and the base variants of them have 12 attention layers instead. We used the same dimension of the hidden layer and the MLP layer for ViT and AiT. 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, Pet, and ImageNet100 datasets, we used a bottleneck size of 512, which selected from a pool of 32.8k/25.1k tokens. For the Triangle dataset, we used a bottleneck size of 64 from a pool of 2.0k tokens. For the relational reasoning tasks, we used a bottleneck size of 256, which selected from a pool of 14.4k tokens. Based on the bottleneck size and the patch pool size, we used 128 memory slots for the Pet and ImageNet100 datasets and 32 memory slots for the other datasets.

Method	Operations		Self-Attention	Low-Rank Memory	Top-Down/Bottom-Up	Bottleneck
	Writing	Reading				
Vision Transformer [2]	×	×	✓	×	BU	×
BlockBERT [28]	×	×	✓	×	BU	✓
BRIMs[39]	×	×	×	×	TD	✓
Modern Hopfield [22]	×	✓	×	✓	BU	×
Perceiver [26]	✓	×	✓	✓	BU	×
Coordination [13]	✓	✓	×	×	BU	✓
Perceiver IO [27]	✓	✓	✓	✓	TD	×
Set Transformer [23]	✓	✓	×	×	BU	×
Luna [25]	✓	✓	×	×	BU	×
GMAT [24]	✓	✓	✓	×	BU	✓
Associative Transformer (Ours)	✓	✓	✓	✓	BU	✓

Table 7. Comparison of attention architectures based on properties of the Global Workspace Theory.

Moreover, we trained the models on the Pet dataset for 300 epochs and on the other datasets for 100 epochs. In relational reasoning tasks, we trained all models for 100 epochs with a batch size of 64.

D. Algorithm

Algorithm 1 Global Workspace Layer

```
**Main program**
**Input:** tokens from the previous layers: $V \in \mathbb{R}^{B \times N \times E}$; learnable low-rank priors: $\gamma \in \mathbb{R}^{M \times D}$
Squash layer concatenates tokens in the batch: $\Xi \in \mathbb{R}^{BN \times E} \leftarrow V \in \mathbb{R}^{B \times N \times E}$
Project $\Xi$ to a latent space with a dimension $D \ll E$: $\Xi W_i^K \in \mathbb{R}^{BN \times D}$
Obtain the attention scores over the projected tokens using priors $\gamma$: $\mathrm{A}_i(\gamma, \Xi) = \mathrm{softmax}(\frac{\gamma(\Xi W_i^K)^T}{\sqrt{D}})$
Tokens compete to write in memory via a bottleneck with capacity $k$: $h_i = \text{top-}k(\mathrm{A}_i)\Xi W^V$ (*Bottleneck Attention Balance Loss is employed for a more diverse token selection)
Update priors $\gamma^t$ with Exponentially Weighted Moving Average: $\hat{\gamma}^{t+1} = (1-\alpha) \cdot \gamma^t + \alpha \cdot \mathrm{LN}(\mathrm{Concat}(h_1, \ldots, h_S)W^O)$
Layer normalization: $\gamma^{t+1} = \frac{\hat{\gamma}^{t+1}}{\sqrt{\sum_{j=1}^{M}(\hat{\gamma}_j^{t+1})^2}}$
Project $\gamma^{t+1} \in \mathbb{R}^{M \times D}$ into a dimension of $E$ as attractors within associative memory: $f_{\mathrm{LT}}(\gamma^{t+1}) \in \mathbb{R}^{M \times E}$
Reconstruct $\Xi$ using attractors $f_{\mathrm{LT}}(\gamma^{t+1})$ with a continuous Hopfield network in one inner step of energy reduction: $\hat{\Xi}^t = \arg\min_{\Xi^t}(-\mathrm{lse}(\beta, f_{\mathrm{LT}}(\gamma^{t+1})\Xi^t) + \frac{1}{2}\Xi^t {\Xi^t}^T + \beta^{-1}\log M + \frac{1}{2}(\max_i |f_{\mathrm{LT}}(\gamma_i^{t+1})|)^2)$
Reshape the tokens into batches as the output of the Global Workspace layer: $\hat{V}^t \in \mathbb{R}^{B \times N \times E} \leftarrow \hat{\Xi}^t \in \mathbb{R}^{(B \times N) \times E}$
**Bottleneck Attention Balance Loss**
Cumulative attention loss: $\ell_{\text{importance}_{i,o}} = \sum_{j=1}^{M} \mathrm{A}_{i,j,o}$
Selected instance loss: $\ell_{\text{loads}_{i,o}} = \sum_{j=1}^{M} (\mathrm{A}_{i,j,o} > 0)$
For each attention head $i$: $\ell_{\text{bottleneck}_i} = \frac{\mathrm{Var}(\{\ell_{\text{importance}_{i,o}}\}_{o=1}^{B \times N})}{(\frac{1}{B \times N}\sum_{o=1}^{B \times N} \ell_{\text{importance}_{i,o}})^2 + \epsilon} + \frac{\mathrm{Var}(\{\ell_{\text{loads}_{i,o}}\}_{o=1}^{B \times N})}{(\frac{1}{B \times N}\sum_{o=1}^{B \times N} \ell_{\text{loads}_{i,o}})^2 + \epsilon}$
Sum the losses over all heads: $\sum_{i=1}^{S} \ell_{\text{bottleneck}_i}$
```

E. Analysis of attention head operating modes

We assume that the competition within the pair-wise attention is important for the model to learn meaningful representations. If such competition exists, a trained model will naturally result in sparser interactions in attention heads. Therefore, we first performed an analysis of the operating modes of different attention heads in a pretrained ViT model by measuring the number of tokens each head is attending to. 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 tokens (Figure 5). The attention sparsity for a specific patch's interactions with other tokens is computed as follows $\arg\min_s \sum_{j=1}^{s} A^{i,j} \geq 0.9$, where $A^{i,j}$ is the attention score allocated to the jth patch by the ith patch. The attention sparsity

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 and ImageNet100	
Patch size	16
Batch size	128
Number of memory slots	128
Bottleneck size	512
Relational reasoning	
Patch size	5
Batch size	64
Number of memory slots	32
Bottleneck size	256

Table 8. Hyperparameters.

score is measured by the minimal number of required tokens whose attention scores add up to 0.90. For instance, there are 65 tokens for the CIFAR-10 task with patch size 4, thus there are 65 interactions for each patch to all the tokens including itself. An attention head has a higher sparsity if the median $\bar{s}$ of the required tokens is smaller, which is the number in the center of each panel. 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 Bidirectional Encoder Representations from Transformers (BERT) for results on the natural language processing (NLP) tasks [22].

Compared to NLP tasks, much less sparsity was found in ViT models. This can be attributed to the long-range dependencies of the attention mechanism were more frequently employed in handling patches of images, compared to handling tokens in text sequences. In ViT models, an input image is divided into non-overlapping patches which are then treated as separate ”tokens” and fed into the transformer model. By contrast, the sequential nature of the NLP text data allows the model to capture long-range dependencies among long-range tokens more easily. Furthermore, depending on the observation of sparsity, the bottleneck size proposed in this work can vary across layers in practice, and different datasets with varying complexities require different capacities as well. When a model has fewer layers, a constant bottleneck size can perform

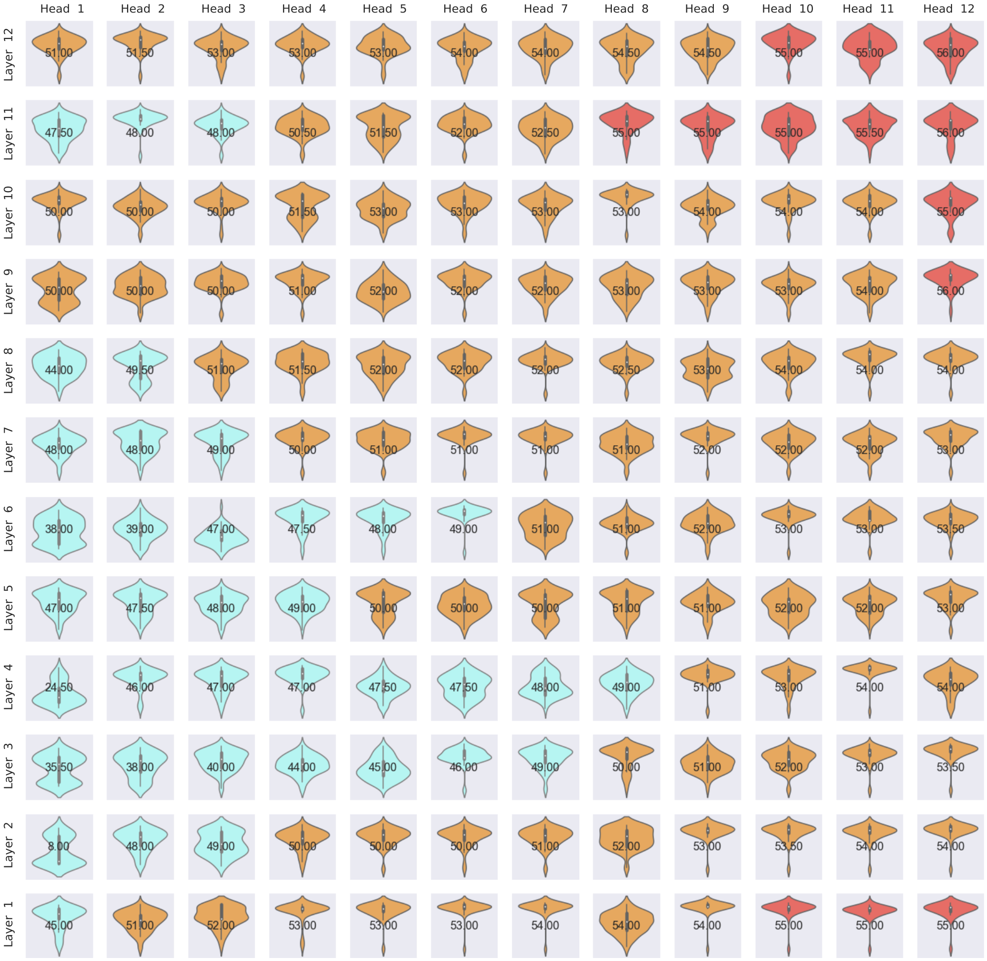


Figure 5. Analysis of operating modes of attention heads in the ViT-Base model. We recognize three different groups of attention heads based on their sparsity scores. Group (I) in light blue: High sparsity heads abundant in the middle layers 3-6. The vast majority of these heads only used 50% or fewer 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 tokens were attended to. The global workspace layer will provide the inductive bias to attend to the essential tokens more effectively.

competitively with a well-tuned one. However, for deeper models, further investigation is needed to determine the optimal capacity size. For example, middle layers might benefit more from a smaller bottleneck compared to other layers.

F. Ablation study during test time

During the test time, we included a new ‘W/O Attention’ ablation, where the multi-head cross-attention is disabled, and only the Hopfield networks are leveraged for the retrieval from the explicit memory. We evaluated ViT performance in both image classification (with AiT-Medium) and relational reasoning tasks (with AiT-Small). Table 9 showed that the multi-head cross-attention with the bottleneck is important for improving AiT's performance during the test time.

Methods	CIFAR10 (%)	CIFAR100 (%)	Triangle (%)	Oxford Pet (%)	Relational tasks (%)
AiT	**84.59**	**60.58**	**99.57**	**30.05**	**76.82**
W/O Attention	84.50	60.56	99.56	28.68	75.17

Table 9. Performance comparison with the W/O Attention ablation.

G. Efficacy of the bottleneck attention balance loss

The Bottleneck Attention Balance Loss facilitates the learning of priors that attend to diverse sets of tokens. We demonstrate the efficacy by visualizing the bottleneck attention scores (Figure 6) and the corresponding patches of the selected tokens by the bottleneck attention (Figure 7). We employed as a metric for patch diversity the ratio of distinct tokens in all the selected tokens.

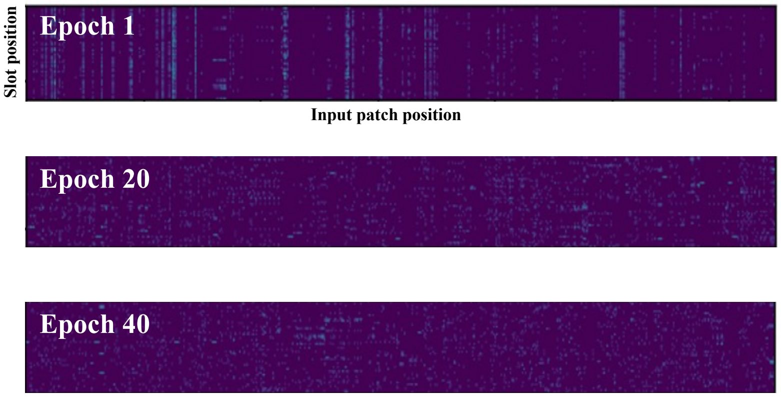


Figure 6. The Bottleneck Attention Balance Loss facilitates the selection of more diverse tokens from various input positions.

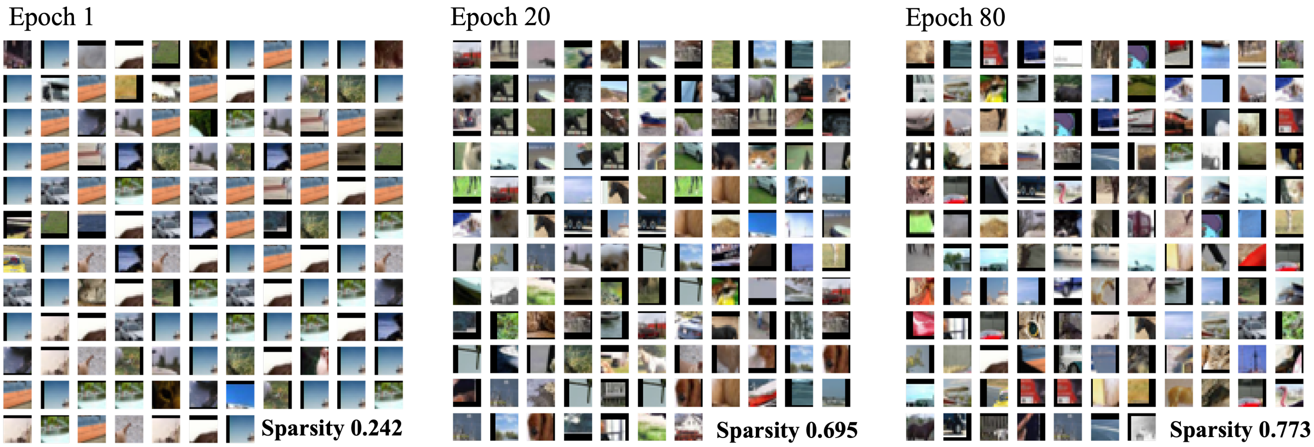
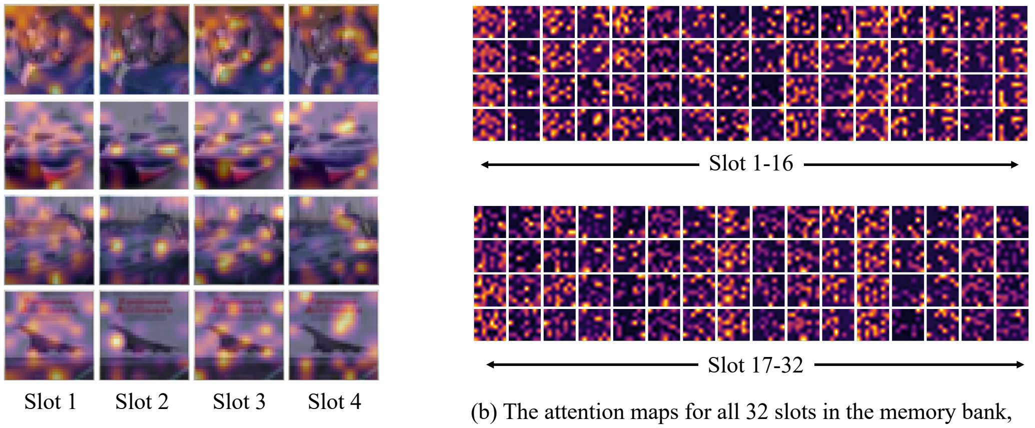


Figure 7. The corresponding patches of the selected tokens by the bottleneck attention in CIFAR-10, demonstrating an apparent increase in the diversity of selected tokens as training progresses.

We visualized the attention scores over different tokens in Figure 8. Despite employing tokens across different batch samples, each prior eventually learned to extract from similar patch input positions, demonstrating emergent spatial specialization.

H. Hopfield networks energy

In traditional Hopfield networks, it is possible to store N samples and retrieve them with partially observed or noisy patterns by updating model weights. During retrieval, these partially observed or noisy patterns converge to one of these attractors, minimizing the Hopfield energy. Unlike traditional Hopfield attractors that incorporate the implicit memory within its model parameters, AiT decouples the memory from the Hopfield network by introducing the learnable explicit memory. This memory serves the functions of both priors in the bottleneck attention and attractors in Hopfield networks. Consequently, the Hopfield network does not need to store different inner states every batch time, instead, we can reuse the learned memory bank from the bottleneck attention to update and maintain a set of attractors with the trainable linear transformation. The



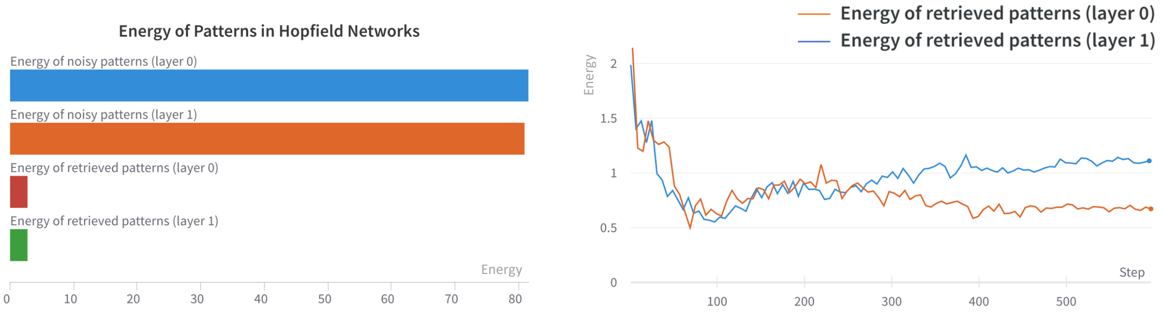
(a) A glimpse of the attention maps at Slot 1 to Slot 4 of four distinct input images.

(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 8. Prior-guided attention visualization highlights selected tokens, with each prior having learned to extract from similar patch input positions across different samples. These priors exhibit emergent specialization, focusing on specific spatial areas in an image (using the first global workspace layer of AiT-Small in CIFAR-10).

proposed architecture is an attractor network in the sense that, in every batch, a pattern converges to one of these attractors derived from the priors stored in the explicit memory bank.

Moreover, the information retrieval 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. All tokens reach their minimum at the same time and the global energy in Equation 5 is guaranteed to decrease. 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. A successful retrieval results in substantial reduction in the state energy (Figure 9).

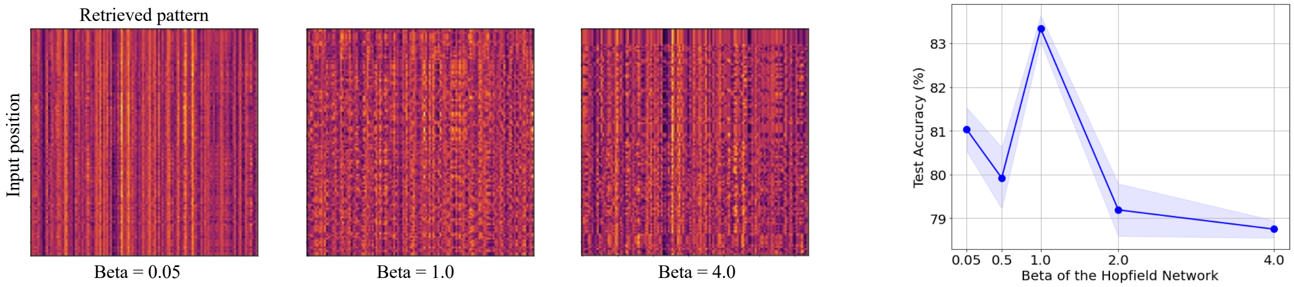


(a) Global energy over the implicit iteration time step

(b) Global energy over the explicit batch time steps

Figure 9. AiT-Small's patch representation energy for the CIFAR-10 task. The Hopfield network operates by iteratively decreasing the energy of an input state with respect to the attractors stored in its memory. This reduction in energy enables the retrieval of a representation that closely aligns with learned attractors, effectively leveraging knowledge within the associative memory. The energy is guaranteed to decrease over the iteration time step for every retrieval. Over the batch time steps, the energy generally decreases, especially during the early stages of training.

Furthermore, we investigated the effect of the inverse temperature β on the information retrieval capability of Hopfield networks in Figure 10. We found that using an inverse temperature of 1.0 obtained the best retrieval performance based on the Hopfield networks. The results suggest that the inverse temperature parameter requires tuning to reach optimal performance. We aim to study a mechanism to adjust β adaptively in a future study.



(a) Retrieved patterns from the Hopfield network for the first 128 patch positions in the input.

(b) Test accuracy when applying varying beta scores.

Figure 10. Varying the inverse temperature score influences the formation of the metastable states that are mixtures of patch representations. A smaller β is more likely to generate such metastable states, while a larger β leads to a stronger separation of different patterns. However, the results showed that a larger β could also lead to local minima, where input patterns were reconstructed to the same pattern.