Agreement aware and dissimilarity oriented GLOM

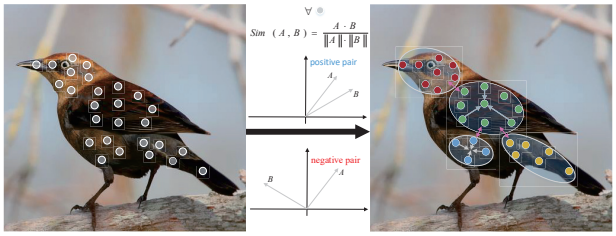
Ru Zeng[1] Yan Song[1]* Yang Zhang[2] Yanling Hu[3] Hui Yu[4]
[1]University of Shanghai for Science and Technology
[2]Shanghai Urban and Rural Construction and Transportation Development Institute
[3]Guangxi Medical University [4]University of Glasgow

Abstract

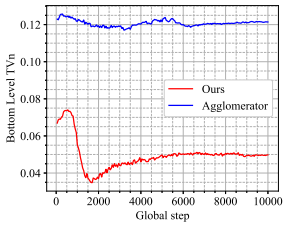
GLOM, an innovative departure from standard deep learning architectures, has been proposed and gained special concern recently due to its good interpretability in representing part-whole relationships in computer vision. However, GLOM faces challenges in achieving agreement and is usually computationally demanding. First, current implementations struggle to produce identical vectors that reliably converge to represent nodes in a parse tree. Second, GLOM is computationally intensive due to the need to maintain equal resolution across all levels. To address these issues, inspired by contrastive learning, we proposed a contrastive agreement enhancer (CAE), which effectively promotes agreement between positive embedding pairs while pushing apart negative pairs, thereby facilitating forming distinct "islands." Furthermore, we introduce a dissimilarity-focused head (H_d) to reduce redundancy in the top-level embeddings, where embedding weights for downsampling are negatively correlated with similarity within a sliding window. The results of comparison experiments indicate that the proposed approach delicately retains informative content and significantly reduces the number of parameters. Additionally, the ablation experiments and visualization results demonstrate that CAE successfully promotes islands of agreement.

1. Introduction

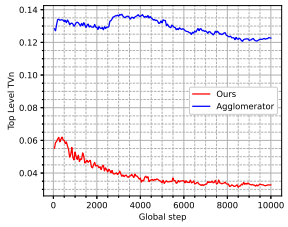
As deep neural networks continue to achieve impressive performance across various tasks, there is increasing focus on model interpretability, particularly in the domain of computer vision [1]. GLOM was proposed to address this need, offering a promising solution to the challenge of representing part-whole hierarchies in neural networks [10]. It integrates the strengths of transformers, neural fields, contrastive representation learning, distillation, and capsules, garnering significant attention upon its introduction. At its core, GLOM uses "islands"—groups of similar feature vectors dynamically identified as coherent units, such as parts of an object—to form nodes in a parse tree. However, current implementations [6, 8] of GLOM struggle to produce similar vectors that reliably reach agreement to form these islands. Moreover, GLOM models are computationally intensive due to the need to maintain equal resolution across



(a) An illustration for the contrastive agreement enhancer



(b) TV_n for bottom level



(c) TV_n for top level

Figure 1. a): To tackle the challenge in current GLOM implementations, where vectors struggle to form islands representing parts and wholes and fail to grow larger islands at higher levels, we introduce a contrastive agreement enhancer. This approach effectively promotes agreement between positive embedding pairs while pushing apart negative pairs, thereby forming distinct islands. By adjusting the threshold for distinguishing positive and negative pairs, we ensure increased smoothness at higher levels that indicate the island grows larger. b) and c): Total variance of normalized embeddings (TV_n) is used as a metric to quantitatively measure smoothness (with lower TV_n indicating greater smoothness). Our method enhances smoothness at both the bottom and top levels, with the top level achieving greater smoothness.

*Corresponding author: sonya@usst.edu.cn

all levels. These limitations inspire the enhancements proposed in this paper, which seek to address the vector agreement problem and improve computational efficiency.

In the GLOM system, embeddings at each level are feature vectors that gradually converge to form distinct islands of nearly identical vectors, representing parts like the head, body, and tail, as shown in the right panel of Fig. 1a. To achieve this, GLOM incorporates a regularizer [6] that minimizes the cosine distance between the bottom-up predictions from lower-level embeddings and the top-down predictions from higher-level embeddings at adjacent levels. However, a practical challenge arises due to the need to balance multiple regularizers between various levels, making it difficult to properly manage their interactions. Another option is a regularizer that encourages similarity between embeddings of nearby locations, but this can lead to the collapse of representations. Although contrastive learning, which commonly uses negative samples to prevent collapse [3], could be a solution, it introduces the challenge of producing distinct islands and recognizing objects, as it relies on finding suitable negative samples to differentiate these islands. Additionally, as the islands grow larger at higher levels to represent whole entities like birds, the embeddings should become more similar, resulting in increased smoothness across the feature map. We use total variance as a metric to quantitatively measure this smoothness. As illustrated by the blue line in Fig. 1b and Fig. 1c, we observe that the total variance [22] of GLOM's normalized embeddings is higher at higher levels than at lower levels, which contradicts GLOM's theoretical expectations.

To address these two issues, we propose a contrastive agreement enhancer (CAE), as shown in Fig. 1a that computes the cosine similarity between two embeddings and determines whether they belong to a positive or negative pair based on a predefined threshold. This mechanism, inspired by contrastive learning, promotes similarity between positive pairs while pushing apart negative pairs. CAE offers several advantages: 1) it can be easily integrated into every level of GLOM; 2) it is not a regularizer, eliminating the need to balance multiple regularizers across levels; 3) it effectively promotes embedding agreement to form distinct islands; and 4) by adjusting the threshold, it ensures increased smoothness at higher levels.

Another focus of this paper is mitigating the computational challenges in GLOM, which arise from maintaining equal resolution across all levels. Specifically, when implementing training or downstream tasks, different heads are customized and added to the top of GLOM. If these heads receive the entire top-level embeddings as input, the computational burden becomes significant because the top-level resolution is as high as the bottom level. A straightforward solution is to downsample the embeddings, based on the premise that redundancy exists within it. Redundancy is indeed present at the top level, as the embeddings there have reached agreement and become highly similar. To handle this, we design a dissimilarity-focused head (H_d), which is integrated at the top of GLOM. H_d similarly to common pooling methods, where a small rectangular window slides across the input feature map. For each window position, H_d calculates the similarity of each embedding with the others within the window. The weights for each embedding are then computed based on the negative correlation of these similarities. The output of H_d is the weighted sum of the embeddings. The advantages of H_d include significantly reducing the parameters and computational load of the heads. Moreover, compared to max pooling, H_d preserves more informative content by focusing on embeddings with lower similarity, which tend to contain more useful information—this is an improvement over max pooling, which just discards all but the highest-value embedding.

Overall, the main contributions of this work can be summarized as follows. 1) A new implementation of GLOM is proposed that aligns more closely with GLOM's theoretical framework for forming hierarchies of grouped embeddings to represent nodes in a parse tree. 2) A novel contrastive agreement enhancer is designed to promote agreement between positive embedding pairs while pushing apart negative pairs, thereby forming distinct "islands." 3) A dissimilarity-focused head is developed to reduce redundancy in the top-level embeddings, preserving more informative content by concentrating on embeddings with lower similarity, which tend to contain more useful information. 4) Quantitative and qualitative analyses show that CAE effectively promotes islands of agreement and H_d contributes greatly to improving model efficiency across datasets.

2. Related work

Capsule networks and GLOM: GLOM provides a more efficient approach to representing part-whole hierarchies than capsule networks (CapsNets) for several reasons [10]. First, CapsNets often face inefficiencies in part representation due to the need to pre-allocate neurons for possible parts at specific locations, leading to underutilized capsules during inference [29, 30]. In contrast, GLOM leverages all neurons to form clusters, facilitating knowledge sharing across locations and improving part representation efficiency. Second, conventional CapsNets require presetting the number of clusters to represent higher-level parts. For instance, studies like [23] and [11] preset the number of capsule types and slice the feature map along the channel direction to form capsules. GLOM, however, allows embeddings to naturally form islands without the need for predefined clusters. Lastly, GLOM eliminates dynamic routing by enabling each part's location to independently construct its own vector representation of the whole, avoiding the need to route information to specific capsules. How-

ever, GLOM is still in its early stages of development and has its own limitations. For example, it primarily functions as an imagery model, with few implementations available to thoroughly evaluate its performance [6, 8]. Current implementations struggle to achieve proper agreement between embeddings, which hampers the formation of islands in GLOM. Additionally, embeddings are set at the same resolution across both low and high levels, leading to redundancy in the model's trainable parameters. This paper focuses on addressing these shortcomings.

Routing and embedding updating: Routing algorithms are a crucial aspect of CapsNets, intended to replace the pooling algorithms typically used in CNNs by assessing the agreement between capsules [23]. This agreement reflects how much lower-level capsules support the existence of a higher-level capsule, based on the similarity of their predictions [11]. Many works about CapsNets focus on reducing the high computation burden in the capsule routing algorithm. For instance, [27] eliminates the iterations in routing and uses capsule distribution variance to activate high-level capsules. Additionally, [2, 4, 17, 19, 20] introduce efficient attention mechanisms into routing, significantly reducing computational overhead. In contrast, GLOM eliminates the need for routing but depends on similar embeddings to iteratively update and form 'islands' representing parts or whole objects [10]. Both routing and embedding updates aim to use similar capsules/embeddings to represent nodes in a sparse tree, organizing objects into a part-whole hierarchy. A key focus of this paper is the role of embedding similarity, which informs the design of the proposed CAE and H_d, each serving different purposes. CAE improves GLOM's island formation by promoting agreement between similar embeddings and disagreement between dissimilar ones, while H_d reduces computational overhead at higher levels by selecting embeddings that are less similar to others, suggesting they may contain more valuable information.

3. Method

Our proposed model aims to efficiently implement the GLOM architecture while more closely aligning with its theoretical framework for creating hierarchical islands of embeddings that represent nodes in a parse tree, with these islands expanding at higher levels. Additionally, the model is designed to be lightweight, facilitating broader potential applications of the GLOM architecture.

Here, we start with the introduction of the mathematical notation needed to explain the details of the main components of the architecture. Given an input image I, it is spatially transformed into $N = h \times w$ patches. The n-th patch corresponds to column $C_n(h, w)$ at the spatial location (h, w), where $n \in \{1, \ldots, N\}$. For each column, it consist of K-level embeddings $f^{k,t}_{(h,w)}, k \in \{1, \ldots, K\}$ at time step $t \in \{1, \ldots, T\}$ with a size of d for each embedding. The subscript (h, w) and superscript t are omitted in subsequent instances for readability. f^k and f^{k+1} are consecutive and represent the part and whole respectively. They are connected by a bottom-up encoder and a top-down decoder which are shared across the spatial dimension.

3.1. Patches embedding

The pipeline is demonstrated in Fig. 2. Firstly, input image I, with the shape of $H \times W \times c$, is transformed into patch embeddings. Following the recipe provided in [16], a convolutional Tokenizer is used to extract feature maps with the size of $H' \times W' \times c'$. Next, it is reshaped to the size of $(h \times r) \times (w \times r) \times c'$, where r is the patch size, $H' = h \times r$, and $W' = w \times r$. Then, we reshape and permute it with the size of $h \times w \times d$, where $d = r \times r \times c'$. At last, the n d-dimensional features act as the embeddings $f^{1,t}$ at the bottom level in columns C_n at time step t.

3.2. Embedding updating

Secondly, we introduce the propagation phase, which governs the interactions between levels and across locations. As shown in Fig. 3, the embedding of a column at location n and level k is updated by integrating multiple sources of information: predictions from lower and higher levels at the same location n using a bottom-up encoder and a top-down decoder, predictions from embeddings at other locations within the same level via self-attention and a contrastive agreement enhancer, and the historical context (i.e., the embedding from the previous iteration). Next, we elaborate on the components of the model that drive this embedding update process.

Bottom-up encoder $\mathcal{N}_{BU}$: Taking a low-level embedding $f^{k,t}$ as input, the bottom-up encoder $\mathcal{N}_{BU}$ is used to predict high-level embedding $f^{k+1,t+1}$ with the same size of d. It is constructed by two fully connected layers and shared across n columns.

Top-down decoder $\mathcal{N}_{TD}$: Given a high-level embedding $f^{k+1,t}$ as input, the top-down decoder $\mathcal{N}_{TD}$ is applied to predict the corresponding low-level embedding $f^{k,t+1}$. The intuition is that if the model understands the whole, it can effectively predict its parts. Like the encoder, $\mathcal{N}_{TD}$ also comprises two fully connected layers.

Self-attention $\mathcal{A}_{SA}$: In GLOM, the interaction between columns is implemented by a simplified self-attention that query, value, and key vector are all the same as input. The aim of self-attention is to produce islands of identical embeddings at a level by making each embedding vector at that level regress towards other similar vectors at nearby locations. $\mathcal{A}_{SA}$ is computed as:

$$\mathcal{A}_{SA}(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d}})V, \tag{1}$$

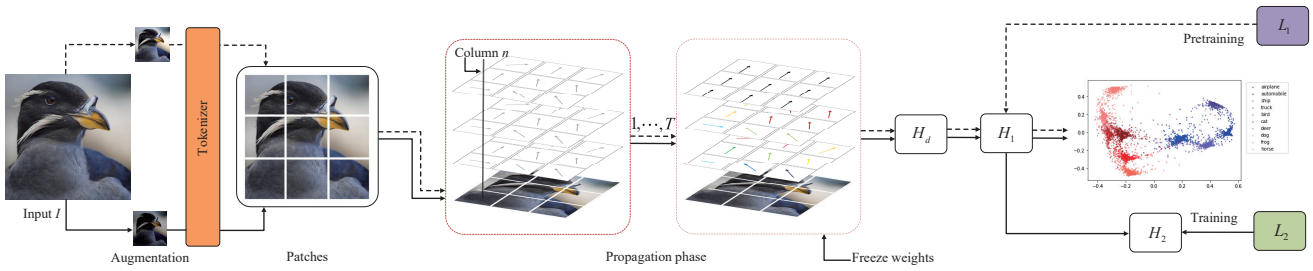


Figure 2. Our model is pre-trained using contrastive learning, followed by supervised training with frozen weights. The input images I are augmented and transformed into patches. These patches are then processed to construct a part-whole hierarchy over T iterations, where embeddings at different levels gradually converge as the levels ascend. Next, the top-level embeddings are fed into H_d, which performs downsampling based on weights that are negatively correlated with similarity. In the subsequent step, the embeddings are sent to H_1 for contrastive pre-training. Finally, the output embeddings are passed to H_2 for supervised training with frozen weights.

where $Q, K, V = f^{k,t}$. It should be noted that $\mathcal{A}_{SA}$ alone is insufficient for creating part-whole hierarchies, as it lacks the flexibility to adjust the degree of agreement (e.g., it struggles to encourage embeddings to achieve greater consensus at higher levels). To address this, GLOM incorporates additional regularization. In contrast, our model adopts a more effective strategy, drawing inspiration from contrastive learning, to promote agreement among embeddings.

Contrastive agreement enhancer $\mathcal{A}_{CAE}$: the function of our proposed CAE is two-fold: enhancing the agreement between positive embedding pairs; and pushing apart the negative embedding pairs. In contrastive learning, positive sample pairs are generated by applying different augmentations to the same image, while negative sample pairs are derived from images belonging to different classes. Unlike contrastive learning, in $\mathcal{A}_{CAE}$, the classification of embedding pairs as positive or negative is based on the similarity between embeddings. To be specific, the similarity is defined as:

$$S = \frac{f_{n1} f_{n2}^T}{\|f_{n1}\| \|f_{n2}\|}, n1, n2 \in \{1, \ldots, N\}, \tag{2}$$

where the size of similarity map S is $N \times N$. For positive and negative embedding pairs, we could set different thresholds. Here we take a simple case for example with $S = 0$ as a threshold for both positive and negative embedding pairs. $S > 0$ means that the angle between embeddings f_{n1} and f_{n2} lies within the range $(0°, 90°)$. In this case, f_{n1} and f_{n2} is a positive embedding pair and could belong to the same island. Conversely, $S < 0$ indicates that f_{n1} and f_{n2} may belong to different islands thereby being a negative pair. This process can be described as:

$$S_p = S \cdot \mathbb{I}(S \geq 0) + (-\infty) \cdot \mathbb{I}(S < 0), \tag{3}$$

$$S_e = S \cdot \mathbb{I}(S < 0), \tag{4}$$

where S_p and S_e represent the similarity map for positive and negative embedding pairs respectively. $\mathbb{I}$ is the indicator function. In the model's bottom level, the threshold is set to 0, while at the top level it is set to -0.2. Next, we compute the embeddings at the time step $t + 1$:

$$f_p^{t+1} = softmax(S_p) f^t, \tag{5}$$

$$f_e^{t+1} = S_e f^t, \tag{6}$$

$$\mathcal{A}_{CAE}(f^t) = f_p^{t+1} - f_e^{t+1}, \tag{7}$$

where f_p^{t+1} are the positive embeddings and f_e^{t+1} are the negative embeddings. Compared with f_e^{t+1}, the computation of f_p^{t+1} evolves the softmax function since the main purpose is to facilitate agreement between embeddings and softmax is good at eliminating the effect of negative embeddings ($S < 0$). In addition, to push apart between positive and negative embeddings, we minus f_p^{t+1} by f_e^{t+1} to predict the f^{t+1}. An illustration for CAE is given in Fig. 1a.

Fusion: At each time step t, $f^{k,t}$ is computed by the fusion of predictions from $\mathcal{N}_{BU}, \mathcal{N}_{TD}, \mathcal{A}_{SA}$, and $\mathcal{A}_{CAE}$:

$$f^{k,t} = w_{bu}\mathcal{N}_{BU}(f^{k-1,t-1}) + w_{td}\mathcal{N}_{TD}(f^{k+1,t-1}) + w_{la}f^{k,t-1} + w_{as}\mathcal{A}_{SA}(f^{k,t-1}) + w_{ac}\mathcal{A}_{CAE}(f^{k,t-1}). \tag{8}$$

where $w_{bu}, w_{td}, w_{la}, w_{as}, w_{ac}$ are trainable parameters.

3.3. Dissimilarity-focused head

With the aim to relieve the computation overhead introduced by which the embeddings at the top level and the lower level have the same resolution, we proposed a dissimilarity-focused head H_d which receives top-level embeddings as input to do downsampling as shown in Fig. 4, and it greatly reduces the number of parameters.

H_d is inspired by the idea that embeddings at the top level should exhibit agreement, which can result in redundancy between them. To quantify this redundancy, we apply indirect thinking and consider similarity as a suitable metric, since the more similar the embeddings are, the more

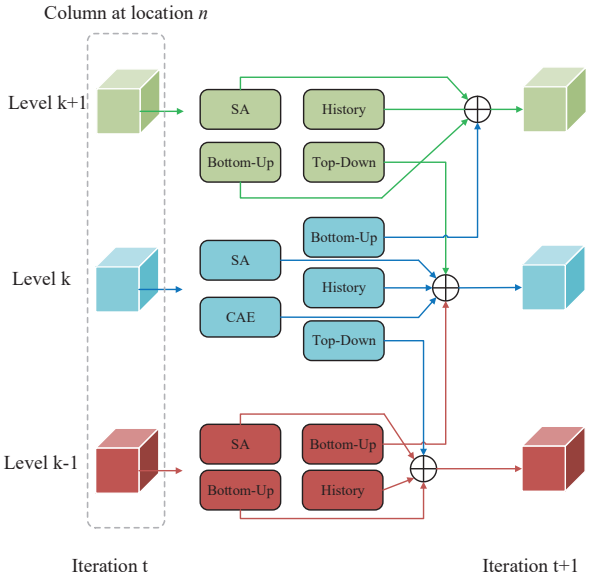


Figure 3. Architecture of model. The column at location n is updating with a fusion of bottom-up and top-down prediction, history, SA, and CAE.

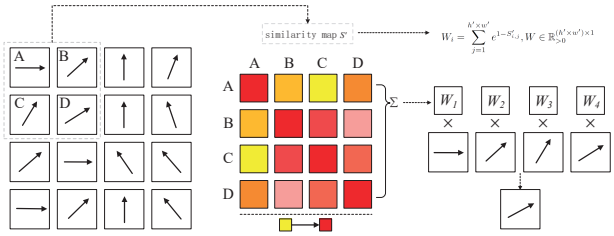


Figure 4. An illustration for the dissimilarity-focused head which is designed to reduce redundancy in the top-level embeddings, where embedding weights for downsampling are negatively correlated with similarity within a sliding window.

redundant they become. To be specific, given the top embeddings $f^K \in \mathbb{R}^{h\times w\times d}$, we first transform them into $p \times p$ patches each with the shape of $h' \times w' \times d$ where $h' = \frac{h}{p}, w' = \frac{w}{p}$. For each patch, similarity map S' with the shape of $(h'\times w')\times(h'\times w')$ is computed using Eq. 2. Next, we compute the weights W for each embedding within the patch:

$$W_i = \sum_{j=1}^{h'\times w'} e^{1-S'_{i,j}}, i, j \in \{1, \ldots, h' \times w'\},$$
$$W = \frac{W}{max(W_i)}, W \in \mathbb{R}^{h'\times w'}_{>0}, \tag{9}$$

where $1 - S'_{i,j}$ means that embeddings with higher similarity with others should be assigned with lower weight to downsample. Besides, the weights are summed to measure the overall weight for embedding f_i. Finally, the downsampled output is computed by the weighted sum within each patch.

$$H_d(f^K) = \sum_{i=1}^{h'\times w'} W_i f_i, \tag{10}$$

where $H_d(f^K) \in \mathbb{R}^{p\times p\times d}$.

3.4. Training

The training of our model follows the setup in [8], consisting of two steps: 1) a pre-training phase using a supervised contrastive loss function, and 2) a classification training phase using cross-entropy loss.

For pre-training, various data augmentation is adopted to each image I within a batch B, producing pairs of image I_a and $I_b, a, b \in \{1, \cdots 2B\}$. Then, the image pairs are fed to our model to produce representation $C_{n,a}$ and $C_{n,b} \in \mathbb{R}^d$, $n \in \{1, \cdots, N\}$ at the top level K. In the following, we reshape them to the shape of $N \times d$ and send them to a contrastive head H_1 which is constructed with two fully connected layers. The output is denoted as O_a and O_b. Finally, a supervised contrastive loss is adopted to pre-train the model:

$$L_1 = -log\frac{e^{sim(O_a,O_b)}}{\sum_{i=1}^{2B}\mathbb{I}_{[i\neq a]}e^{sim(O_a,O_i)}}, \tag{11}$$

where $\mathbb{I}_{[i\neq a]}$ is an indicator function that values 0 if i and a belong to the same image class, and 0 otherwise. sim is a cosine similarity function that measures the similarity of normalized input.

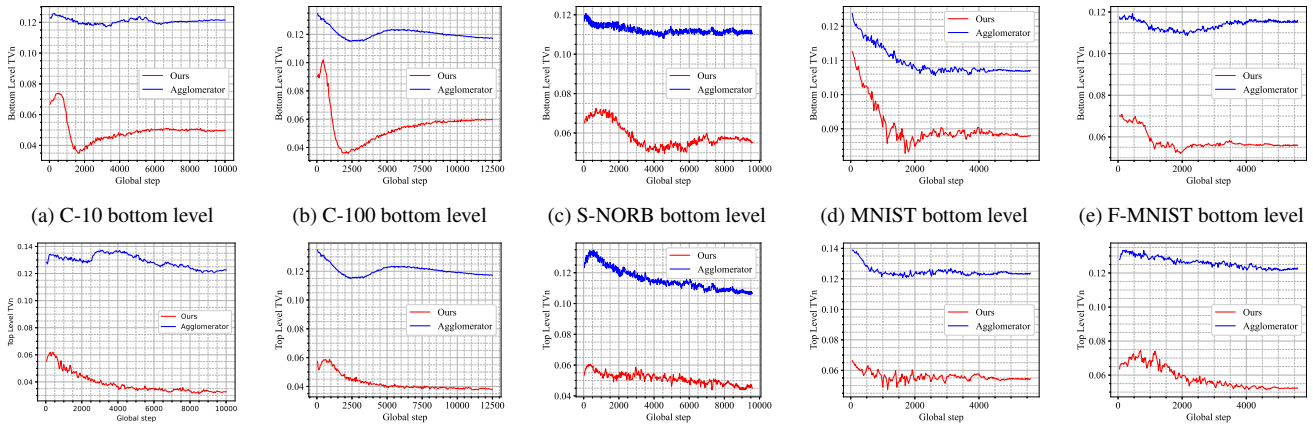
After the model is pre-trained with contrastive loss, we freeze the network weights. Besides, we add a classification head H_2 on the top of contrastive head H_1. The H_2 is a linear layer that projects the representation to the final out with the dimension of C. C is the number of classes. The cross-entropy loss is given as:

$$L_2 = -\frac{1}{C}\sum_{i=1}^{C} p_i log q_i, \tag{12}$$

where p_i is the true probability distribution for class i and q_i is the predicted probability for class i.

4. Experiment

Our experiments encompass quantitative and qualitative analyses. Quantitatively, we evaluate the formation of islands of agreement using the total variance of normalized embeddings and compare model performance across multiple datasets. An ablation study further examines the contributions of the proposed CAE and H_d. For the qualitative



(a) C-10 bottom level (b) C-100 bottom level (c) S-NORB bottom level (d) MNIST bottom level (e) F-MNIST bottom level

(f) C-10 top level (g) C-100 top level (h) S-NORB top level (i) MNIST top level (j) F-MNIST top level

Figure 5. The comparison of our model with that of the Agglomerator in terms of TV_n at both the bottom and top levels across five datasets. It is evident that the total variance of our model is considerably lower than that of the Agglomerator, indicating that the contrastive attention mechanism effectively improves agreement in representing parts, which is essential for the theory of GLOM. Furthermore, the TV_n of our model is greater at the bottom level than at the top level, suggesting that agreement tends to increase as the levels rise.

analysis, we visualize the vector representations of emerging islands across levels. To illustrate CAE's enhancement of agreement intuitively, we depict representations with arrows spanning various locations and levels. Experiments are conducted on the CIFAR-10, CIFAR-100 [13], MNIST [14], FashionMNIST [26], and SmallNORB datasets [15]. The details of the training settings are provided in the supplementary material. The source code is publicly available at: https://github.com/zengru001usst/GLOM

4.1. Agreement Evaluation

To verify the effectiveness of the proposed CAE, total variance [22] of the normalized embeddings (TV_n), as shown in Eq. 13, is employed as a metric to quantitatively measure the smoothness of embeddings at specific levels within the models. The embeddings are normalized, as a trivial solution may occur where total variance is minimized by making every $f_{i,j}$ infinitesimal. Lower TV_n indicates greater smoothness, meaning that more embeddings are similar and achieve a higher level of agreement.

$$
\begin{aligned}
TV_n &= \frac{1}{N}\frac{1}{d}\sum_{i=1}^{N}\sum_{j=1}^{d}(f'_{i,j}-\mu_j)^2, \\
\mu_j &= \frac{1}{N}\sum_{i=1}^{N} f'_{i,j}, \\
f'_{i,j} &= \frac{f_{i,j}}{\sqrt{\sum_j f_{i,j}^2}}.
\end{aligned} \tag{13}
$$

The results presented in Fig. 5 compare the TV_n of our model with that of the Agglomerator at both the bottom and top levels across five datasets. It can be observed that the TV_n of our model is significantly lower than that of the Agglomerator, indicating that the CAE effectively enhances agreement in representing parts, which is crucial for the theory of GLOM. Additionally, the TV_n of our model is higher at the bottom level compared to the top level, suggesting that agreement increases as the levels ascend. This is consistent with GLOM's framework, where embeddings at lower levels gradually cluster locally to represent parts, while embeddings at higher levels cluster globally to represent the whole. In contrast, the Agglomerator fails to maintain this characteristic.

4.2. Classification Results

To demonstrate the efficiency of the proposed model, we report the classification results for each dataset in terms of error percentage and the number of trainable parameters, as shown in Tab. 1. While Capsule Networks (CapsNets), such as [11, 18, 19, 21, 23, 28], perform well on simpler datasets like smallNORB, MNIST, and FashionMNIST, they struggle to scale to more complex datasets with a higher number of classes, such as CIFAR-100, due to inefficiencies in object representation [10]. Convolutional models, such as [9, 24], maintain strong performance across various datasets but fall short in interpretability. Transformer-based and MLP-based methods achieve state-of-the-art performance on more complex datasets, but they lack testing on smaller datasets. As shown in Tab. 1, our model achieves comparable classification error percentages to CapsNets and Agglomerator [8] on simpler datasets, but with significantly fewer parameters, especially compared to Agglomerator. Moreover, our model requires less training time and has a

Method	Ref	Backbone	Errors %					No. params (Millions)	Training Arch.
			S-NORB	MNIST	F-MNIST	C-10	C-100		
E-CapsNet	[19]	Caps	2.54	0.26	-	-	-	0.2	GPU
CapsNet	[23]		2.70	**0.25**	6.38	10.6	82.00	6.8	GPU
Matrix-CapsNet	[11]		1.40	0.44	6.14	11.9	-	0.3	GPU
Capsule VB	[21]		1.60	0.30	5.20	11.2	-	0.2	GPU
Res-CapsNet	[18]		0.55	0.55	-	7.62	-	16.4	GPU
IAR-CapsNet	[28]		1.36	0.77	-	-	-	0.18	GPU
ResNet-110	[9]	Conv	-	2.10	5.10	6.41*	27.76*	1.7	GPU
VGG	[24]		-	0.32	6.50	7.74*	28.05*	20	GPU
ViT-L/16	[7]	Transf	-	-	-	**0.85***	**6.75***	632	TPU
ConvMLP-L	[16]	Conv/MLP	-	-	-	1.40*	11.40*	43	TPU
MLP-Mixer-L/16	[25]	MLP	-	-	-	1.66*	-	207	TPU
Agglomerator	[8]	Conv/MLP-/Caps	0.01	0.30	7.43	11.15	40.97	72	GPU
Ours	-		**0.01**	0.30	**5.10**	10.90	41.31	2	GPU

Table 1. Classification results on SmallNORB (S-NORB), MNIST, FashionMNIST (F-MNIST), CIFAR-10 (C-10), and CIFAR-100 (C-100) datasets, demonstrating the efficiency of the proposed model. Results marked with * indicate networks pre-trained on ImageNet.

	H_d	CAE	Errors (%)	No. Param. (Millions)
Baseline	×	×	9.1	72
Model1	×	√	9.0	72
Model2	√	×	11.6	2
Model3	√	√	10.9	2

Table 2. Ablation for dissimilarity-focused head (H_d) and contrastive agreement enhancer (CAE) on CIFAR10.

	Errors (%)	TV_n	No. Param. (Millions)
Average pooling	11.1	0.372	2
Max pooling	11.1	0.473	2
H_d	10.9	0.544	2

Table 3. Comparison of dissimilarity-focused head (H_d), average pooling and maxpooling on CIFAR10.

much smaller architecture compared to most transformer-based and MLP-based methods.

4.3. Quantitative Ablation

To study the contribution of the different components of our model to its performance, we conduct an ablation experiment where we incrementally added the dissimilarity-focused head (H_d) and contrastive agreement enhancer (CAE) to the baseline model. The baseline model is defined by removing both H_d and CAE from our full model. We then compare the classification results and the number of parameters on CIFAR10 to demonstrate the contribution of H_d and CAE to model performance.

The results in Tab. 2 indicate that H_d can significantly reduce the number of trainable parameters in the model while only causing a slight increase in errors, which suggests the presence of redundancy at the top level. Additionally, when H_d is employed, CAM can enhance model performance, as much of the redundancy has been eliminated, allowing CAM to provide valuable supplementary information.

To evaluate the impact of H_d on classification accuracy and TV_n, we compare our model against two alternative downsampling strategies: max pooling and average pooling, by substituting H_d with these methods. Results in Tab. 3 show that the model with H_d achieves a lower error rate and higher TV_n than these alternatives under identical downsampling conditions. The higher TV_n with H_d stems from its emphasis on embeddings with lower similarity, which carry more informative features and enhance classification accuracy.

4.4. Qualitative Ablation

We also conduct two qualitative ablation study to demonstrate the effect of the contrastive agreement enhancer.

The emerging island of agreement. To illustrate the emergence of islands across different levels, the embeddings from various k levels of CIFAR-10 are visualized in Fig. 6. When examining the embeddings at the highest level ($k = 5$), those generated by the model with CAE reveal finer details and a more distinct separation between objects and the background. Furthermore, a comparison of embeddings across different levels demonstrates that as the levels increase, the embeddings progressively reach agreement, effectively capturing part-whole hierarchies.

The representation visualization using arrows. To provide an intuitive illustration of the enhancement in

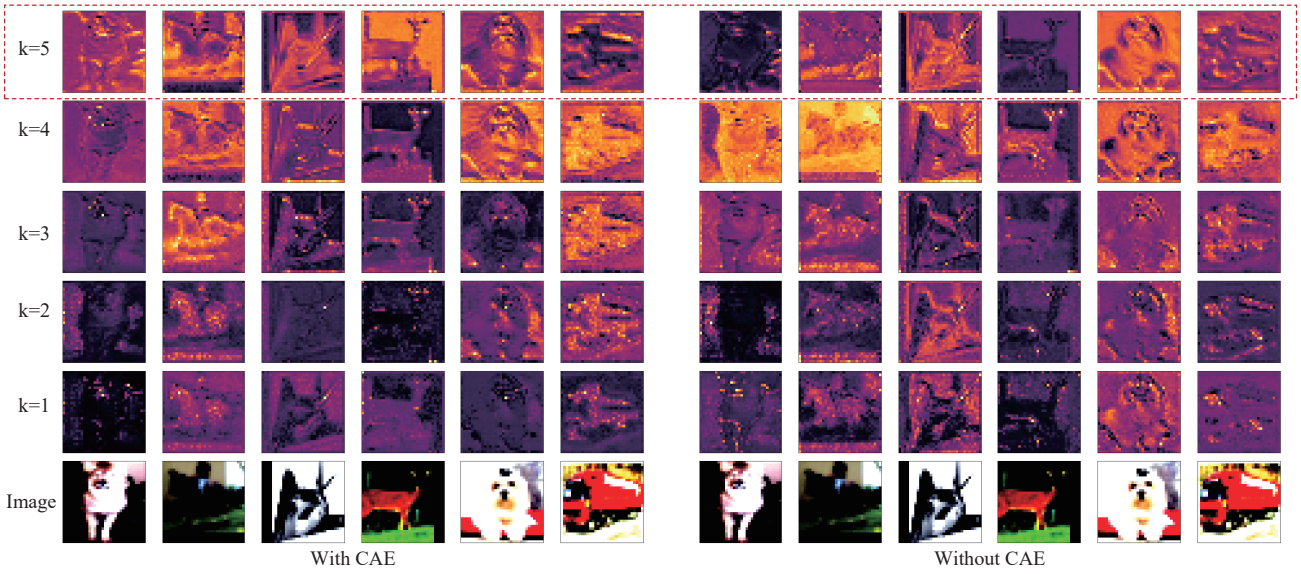


Figure 6. The emerging island of agreement at various k levels from CIFAR-10. At the top level (k=5), the representations generated by the model with CAE display enhanced detail, highlighting a more distinct separation between objects and the background.

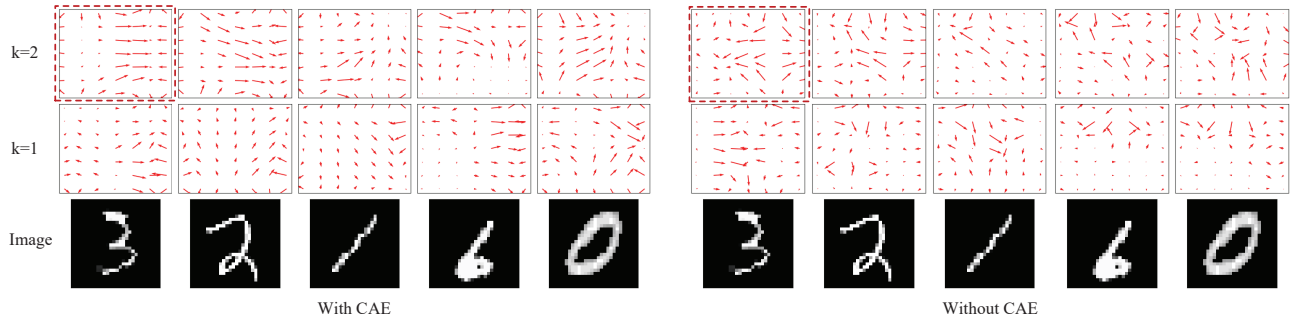


Figure 7. The representation visualization using arrows for model with and without CAE. The direction and length of each arrow represent the gradient direction and magnitude of an embedding at a specific location, computed via the Sobel operator.

agreement introduced by CAE, we visualize the representations using arrows across various locations and levels. The direction and length of each arrow indicate the gradient direction and magnitude of an embedding at a specific location, computed via the Sobel operator. MNIST is selected for visualization due to its simplicity, minimizing potential interference from complex image backgrounds. As depicted in Fig. 7, the arrows in the model incorporating CAE demonstrate greater consistency, whereas those in the model without CAE appear more disordered and inconsistent.

5. Conclusion

In this paper, we have proposed a new implementation of GLOM that aligns more closely with its theoretical framework for forming hierarchies of grouped embeddings to represent nodes in a parse tree. To foster the formation of distinct islands, we have designed a novel contrastive agreement enhancer that encourages agreement between positive embedding pairs while separating negative pairs. To reduce the computational burden of GLOM, we developed a dissimilarity-focused head that preserves more informative content by prioritizing embeddings with lower similarity, which often contain richer information. Both quantitative and qualitative analyses demonstrate that the contrastive agreement enhancer effectively promotes islands of agreement, while the dissimilarity-focused head enhances model efficiency across various datasets. This work significantly improves the practicality of GLOM, and we hope it inspires other researchers to further explore this direction to advance its development.

References

[1] Yoshua Bengio, Yann Lecun, and Geoffrey Hinton. Deep learning for ai. *Communications of the ACM*, 64(7):58–65, 2021. Publisher Copyright: © 2021 Owner/Author. 1

[2] Ran Chen, Hao Shen, Zhong-Qiu Zhao, Yi Yang, and Zhao Zhang. Global routing between capsules. *Pattern Recognition*, 148:110142, 2024. 3

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 2

[4] Jaewoong Choi, Hyun Seo, Suii Im, and Myungjoo Kang. Attention routing between capsules. In *Proceedings of the IEEE/CVF international conference on computer vision workshops*, pages 0–0, 2019. 3

[5] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 113–123, 2019. 1

[6] Laura Culp, Sara Sabour, and Geoffrey E Hinton. Testing glom's ability to infer wholes from ambiguous parts. *arXiv preprint arXiv:2211.16564*, 2022. 1, 2, 3

[7] Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 7

[8] Nicola Garau, Niccoló Bisagno, Zeno Sambugaro, and Nicola Conci. Interpretable part-whole hierarchies and conceptual-semantic relationships in neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13689–13698, 2022. 1, 3, 5, 6, 7

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 6, 7

[10] Geoffrey Hinton. How to represent part-whole hierarchies in a neural network. *Neural Computation*, 35(3):413–452, 2023. 1, 2, 3, 6

[11] Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. Matrix capsules with em routing. In *International conference on learning representations*, 2018. 2, 3, 6, 7

[12] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933. 1

[13] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[14] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 6

[15] Yann LeCun, Fu Jie Huang, and Leon Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, pages II–104. IEEE, 2004. 6

[16] Jiachen Li, Ali Hassani, Steven Walton, and Humphrey Shi. Convmlp: Hierarchical convolutional mlps for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6307–6316, 2023. 3, 7

[17] Yi Liu, Dingwen Zhang, Qiang Zhang, and Jungong Han. Part-object relational visual saliency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(7):3688–3704, 2022. 3

[18] Yi Liu, De Cheng, Dingwen Zhang, Shoukun Xu, and Jungong Han. Capsule networks with residual pose routing. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2024. 6, 7

[19] Vittorio Mazzia, Francesco Salvetti, and Marcello Chiaberge. Efficient-capsnet: Capsule network with self-attention routing. *Scientific reports*, 11(1):14634, 2021. 3, 6, 7

[20] Rita Pucci, Christian Micheloni, and Niki Martinel. Self-attention agreement among capsules. In *Proceedings of the ieee/cvf international conference on computer vision*, pages 272–280, 2021. 3

[21] Fabio De Sousa Ribeiro, Georgios Leontidis, and Stefanos Kollias. Capsule routing via variational bayes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 3749–3756, 2020. 6, 7

[22] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 2, 6

[23] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. *Advances in neural information processing systems*, 30, 2017. 2, 3, 6, 7

[24] Karen Simonyan. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6, 7

[25] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021. 7

[26] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017. 6

[27] Ru Zeng and Yan Song. A fast routing capsule network with improved dense blocks. *IEEE Transactions on Industrial Informatics*, 18(7):4383–4392, 2021. 3

[28] Ru Zeng, Yuzhang Qin, and Yan Song. A non-iterative capsule network with interdependent agreement routing. *Expert Systems with Applications*, 238:122284, 2024. 6, 7

[29] Ru Zeng, Yan Song, and Yuzhang Qin. Spatial attention-based capsule networks with guaranteed group equivariance. *IEEE Transactions on Automation Science and Engineering*, 2024. 2

[30] Ru Zeng, Yan Song, and Yanjiu Zhong. An interpretable unsupervised capsule network via comprehensive contrastive learning and two-stage training. *Pattern Recognition*, 158: 111059, 2025. 2