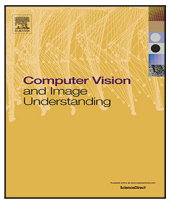
Computer Vision and Image Understanding 249 (2024) 104180

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Spatial attention for human-centric visual understanding: An Information Bottleneck method

Qiuxia Lai [a], Yongwei Nie [b,*], Yu Li [c], Hanqiu Sun [d], Qiang Xu [e]

[a] State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, 100024, China
[b] School of Computer Science and Engineering, South China University of Technology, Guangzhou, 510641, China
[c] School of Computer Science and Technology, Harbin Institute of Technology (Shen Zhen), Shenzhen, 518055, China
[d] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China
[e] Department of Computer Science and Engineering, The Chinese University of Hong Kong, 999077, Hong Kong, China

ARTICLE INFO

Dataset link: https://github.com/ashleylqx/AIB-Ex.git

Keywords:
Information bottleneck
Attention mechanism
Deep learning

ABSTRACT

The selective visual attention mechanism in the Human Visual System (HVS) restricts the amount of information that reaches human visual awareness, allowing the brain to perceive high-fidelity natural scenes in real-time with limited computational cost. This selectivity acts as an "Information Bottleneck (IB)" that balances information compression and predictive accuracy. However, such information constraints are rarely explored in the attention mechanism for deep neural networks (DNNs). This paper introduces an IB-inspired spatial attention module for DNNs, which generates an attention map by minimizing the mutual information (MI) between the attentive content and the input while maximizing that between the attentive content and the output. We develop this IB-inspired attention mechanism based on a novel graphical model and explore various implementations of the framework. We show that our approach can yield attention maps that neatly highlight the regions of interest while suppressing the backgrounds, and are interpretable for the decision-making of the DNNs. To validate the effectiveness of the proposed IB-inspired attention mechanism, we apply it to various computer vision tasks including image classification, fine-grained recognition, cross-domain classification, semantic segmentation, and object detection. Extensive experiments demonstrate that it bootstraps standard DNN structures quantitatively and qualitatively for these tasks.

1. Introduction

Human beings can process vast amounts of visual information in parallel through the visual system (Koch et al., 2006) because the attention mechanism of the Human Visual System (HVS) can selectively attend to the most informative visual stimuli rather than the whole scene (Eriksen and Hoffman, 1972). Recently, this principle of selective attention has been integrated into Deep Neural Networks (DNNs), allowing these systems to focus on task-relevant parts of the input automatically. The incorporation of the attention mechanism has benefited a wide range of computer vision tasks such as semantic segmentation (Wang et al., 2019a; Alcazar et al., 2021; Zhou et al., 2022; Hui et al., 2023), image inpainting (Qin et al., 2021; Chen et al., 2024), salient object detection (Wang et al., 2019c), video object segmentation (Wang et al., 2019b), and visual–language navigation (Chen et al., 2022; An et al., 2024; Gao et al., 2023) with enhanced performance and interpretability.

The attention modules in DNNs can be generally categorized into channel-wise attention and spatial attention. The former learns channel-wise attention scores to modulate the feature maps (Hu et al., 2018), whereas the latter learns spatially-aware attention scores and adjusts the features accordingly (Simonyan et al., 2013). This paper focuses on spatial attention mechanisms, as channel-wise attention would inevitably lose spatial information that is essential for localizing the important parts. Spatial attention can be further divided into query-based and module-based categories. Query-based attention, such as "self-attention" (Vaswani et al., 2017; Ren et al., 2022), generates the attention scores based on the similarity/compatibility between the query and the key content. While having facilitated various computer vision tasks, such dense relation measurements incur heavy computational costs (Han et al., 2020), limiting its practical applications. In contrast, module-based attention uses a trainable network module to directly produce an attention map from an image or feature, offering

* Corresponding author.
E-mail address: nieyongwei@scut.edu.cn (Y. Nie).

https://doi.org/10.1016/j.cviu.2024.104180
Received 15 April 2024; Received in revised form 15 August 2024; Accepted 13 September 2024
Available online 24 September 2024

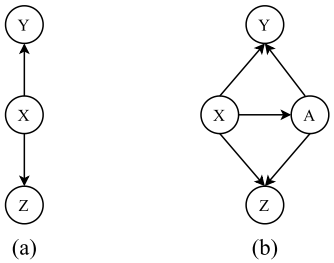


Fig. 1. Graphical model of (a) a probabilistic neural network studied in DVIB (Alemi et al., 2017) and (b) a probabilistic neural network with IB-inspired spatial attention mechanism (*c.f.* Section 3.2).

a more efficient, end-to-end inference process compared with query-based attention. This method has proven effective across multiple computer vision tasks. In this paper, we focus on the module-based spatial attention mechanism and attempt to enhance it using the "Information Bottleneck" theory (Tishby et al., 1999) to advance human-centric visual understanding in DNNs.

The basic idea explored in this paper is that the existing attention mechanisms employed in current DNNs, though successfully simulating aspects of the attention function of HVS, are not explicitly optimized to account for the inherent information processing constraints of the attention mechanism of HVS. For HVS, besides focusing only on important information, it also restricts the amount of information that reaches human visual awareness, such that the brain can process high-fidelity natural scenes in real-time. In contrast, current attention mechanisms trained for certain tasks only measure the relative importance of each spatial location in the feature map to improve task performance. Consequently, the resultant attention maps lack selectivity, failing to ensure that the information bypassed by the attention maps is of minimal redundancy, and meanwhile being sufficient for the task.

To explicitly incorporate the information constraint property of HVS into the attention learning of DNNs, we propose an end-to-end trainable spatial attention mechanism inspired by the "Information Bottleneck (IB)" theory (Tishby et al., 1999). IB theory is a variational principle that describes a tradeoff between preserving meaningful information and compressing the representation (Tishby et al., 1999). Recently, Alemi et al. (2017) realized the IB theory by DNNs for representation learning. Basically, given an input source X and the target Y, IB theory requires a network to learn latent representation Z that maximizes the objective $I(Z;Y)-\beta I(Z;X)$, with X, Y and Z satisfying the joint distribution denoted in Fig. 1(a). This objective aims to minimize the mutual information (MI) between X and Z (denoted by $I(Z;X)$), and maximize the MI between Y and Z (denoted by $I(Z;Y)$). From X to Z, the overall information is compressed as much as possible, while from Z to Y the meaningful information is preserved as much as possible, making Z to work as an "Information Bottleneck" which only allows the most important information to pass.

Our IB-inspired attention mechanism shares insights with Alemi et al. (2017). Specifically, besides X, Y, and Z, we introduce a new random variable A which stands for an attention score map. We devise the new relationships among the four random variables as shown by the graphical model in Fig. 1(b), based on which we impose the IB objective function $I(Z;Y)-\beta I(Z;X,A)$ on the optimization of the attention map A. Compared with existing attention mechanisms, our IB-objective constrained attention mechanism is more aggressive to filter out task-irrelevant information, forcing the attention maps to focus on the most important information. The whole framework is derived from the information-theoretic argument based on the IB principle, resulting in variational attention maps. To further restrict the information bypassed by the attention map, an adaptive quantization module is incorporated to round the attention score to the nearest anchor value. In this way, previous continuous attention values are replaced by a finite number of anchor values, which further confines the information filtered by the attention maps.

A conference version of this paper was presented in Lai et al. (2021), where the basic idea of IB-inspired spatial attention learning was introduced but the effectiveness of the proposed attention mechanism was not sufficiently evaluated. In this paper, we extend the previous version by realizing the key components of the proposed attention mechanism in a wider range of ways and evaluating them on more applications and tasks. The improvements include:

- In the conference paper, we just investigated scaling with a residual-based method to modulate the feature with the attention map. In this paper, more attentive feature modulation strategies are explored, including scaling, biasing, and concatenation.
- Previously, we used Gaussian encoder to calculate the KL divergence between the latent and the prior distributions in closed form. By introducing an adversarial learning strategy (Makhzani et al., 2016), we obviate the need for a tractable latent density, and extend the encoders to two other stochastic encoders (Belghazi et al., 2018).
- Besides visual recognition tasks, we further validate the proposed method on other two computer vision tasks, namely, semantic segmentation and object detection, to assess the generalization ability of the proposed spatial attention mechanism.

In summary, our contributions are:

- We propose an IB-inspired spatial attention learning framework for human-centric visual understanding, which yields variational attention maps that minimize the MI between the attention-modulated representation and the input while maximizing the MI between the attention-modulated representation and the task label. To further filter out irrelevant information, we design a learnable quantization module to round the continuous attention scores to several learnable anchor values.
- We demonstrate the robustness of the proposed IB-inspired attention mechanism w.r.t. different realization strategies for which our method consistently shows advantages compared to the baselines.
- We successfully apply the proposed IB-inspired spatial attention mechanism to lots of tasks in visual recognition and dense prediction, showing its generalization ability across different applications.

2. Related work

In this section, we first review various approaches that incorporate spatial attention within DNNs. Then, we introduce methods that utilize the IB principle to create masks to improve model performance or interpret model behavior.

2.1. Spatial attention mechanism in DNNs

Attention mechanisms have achieved noticeable success in sequence modeling tasks such as speech recognition (Chorowski et al., 2015), machine translation (Bahdanau et al., 2015), and image captioning (Xu et al., 2015). More recently, they have also proven beneficial to a broad spectrum of computer vision tasks, enhancing both the performance and the interpretability of general CNNs. The attention modules in CNNs can be categorized into two broad categories, namely channel-wise attention and spatial attention. The former adjusts feature maps by learning and applying channel-specific weights (Hu et al., 2018), while the latter learns a probabilistic map over the input to enhance or suppress each 2D location according to its relevance to the target task (Simonyan et al., 2013). In this section, we mainly focus on spatial attention modules. For a more detailed exploration of all attention types in computer vision, please refer to Guo et al. (2021).

Query-based/Self-attention is originally developed for query-based tasks (Bahdanau et al., 2015; Xu et al., 2015). This kind of attention is generated by assessing the similarity or compatibility between the query and the key content. For instance, Seo et al. (2018) employ a one-hot encoding of the label to query the image and produce progressive attention for attribute prediction. Jetley et al. (2018) use the learned global representation of the input image as a query and calculate its compatibility with local representation from each 2D spatial location to generate the attention map. Hu et al. (2019) adaptively determine the aggregation weights by considering the compositional relationship of visual elements in local areas. Query-based attention facilitates dense relational mapping in the space, enhancing the discriminative ability of CNNs. However, the substantial computational demands restrict its application to low-dimensional inputs, and typically necessitate significant downsampling of the original images.

Module-based attention. This type of spatial attention map can be directly learned through a network module with a softmax or a sigmoid activation function, which takes an image or feature as input and outputs an attention map. Owing to its effectiveness and efficiency, module-based attention has been widely used in computer vision tasks such as image classification (Woo et al., 2018) and action recognition (Sharma et al., 2015). Our work belongs to this line of research and directly generates a spatial attention map using a network module. Previous efforts focus on the relations among non-local or local contexts to measure the relative importance of each location and overlook the information in the feature filtered by the attention maps. In contrast, our approach takes inspiration from the IB theory (Tishby et al., 1999) to achieve a balance between information compression and prediction accuracy. We propose to learn spatial attention that minimizes the MI between the masked feature and the input, while maximizing the MI between the masked feature and the task label. Our approach can effectively eliminate the redundant information from the input features compared with conventional methods that focus solely on relative importance learning.

2.2. IB-inspired mask generation

IB theory has been explored in tasks such as representation learning (Alemi et al., 2017; Achille and Soatto, 2018) and graph learning (Sun et al., 2022; Yuan et al., 2024). Several applications that incorporate IB theory also generate additive (Schulz et al., 2019) or multiplicative masks or attention maps (Achille and Soatto, 2018; Taghanaki et al., 2019; Zhmoginov et al., 2019) to restrict the information that flows to the subsequent layers, achieved by optimizing the IB objective (Tishby et al., 1999). For instance, Information Dropout (Achille and Soatto, 2018) introduces a generalized dropout operation with information constraints that multiplies the layer feature with a learnable noise mask, thereby regulating the flow of information. This approach essentially optimizes a modified cost function that aligns with the IB Lagrangian of Tishby et al. (1999), fostering the learning of representations that are sufficient, minimal, and invariant for classification tasks. InfoMask (Taghanaki et al., 2019) applies the masks optimized based on IB theory to filter out irrelevant background signals, which enhances the precision of chest disease localization. Zhmoginov et al. (2019) develop IB-inspired Boolean attention masks for image classification in a semi-supervised setting, which completely block the propagation of any information from the masked-out pixels to the model output. Schulz et al. (2019) leverage the IB concept to interpret the decision-making of a pre-trained neural network by adding noise to intermediate activation maps to restrict and quantify the flow of information. The intensity of the noise is optimized to minimize the information flow while maximizing the classification accuracy.

The fundamental distinction between our IB-inspired attention method and mask generation approaches described previously lies in the optimization objectives. Our method incorporates the attention variable A into the joint distribution, whereas the masks generated by earlier methods are typically integrated with either X or Z to fit the vanilla IB objective without being treated as standalone entities as in our approach. Consequently, our strategy explicitly optimizes the attention to adhere to the IB constraints, which differentiates it from previous literature.

3. Methodology

Intuitive Explanation. Our framework is designed based on the Information Bottleneck (IB) principle, which aims to balance compressing input data and preserving relevant information for the task at hand. We extend the IB principle by introducing an additional random variable A, which represents the attention. Our goal is to maximize the resulting AIB objective $I(Z;Y)-\beta I(Z;X,A)$. Similar to Alemi et al. (2017), we derive a framework from the lower bound of this new Attentive IB (AIB) objective. The framework, comprising an attention module, an encoder, and a decoder, is trained using loss functions derived from the AIB objective to ensure that the attention maps produced by the model maximize the mutual information between the latent encoding Z and the output Y, while minimizing that between the latent encoding Z and the combined input X and attention A. This attention mechanism contributes to effective data compression and information preservation.

Since the derivation of the proposed IB-inspired spatial attention mechanism shares similar insights with the DNN realization (Alemi et al., 2017) of the IB principle (Tishby et al., 1999), we briefly review (Alemi et al., 2017) in Section 3.1. Our IB-inspired attention mechanism and the attention score quantization strategy are introduced in Section 3.2. Finally, we show various strategies for optimizing the variational bounds induced by the IB-inspired attention mechanism and discuss the computer vision tasks where our attention method enhances performance.

3.1. Information bottleneck principle

Information bottleneck (IB) is a variational principle for representation learning that considers the tradeoff between meaningful information preservation and representation compression (Tishby et al., 1999). Recently, Alemi et al. (2017) parameterize IB using DNNs. Assume that random variables X, Y and Z follow the joint distribution denoted as in Fig. 1(a), *i.e.*, $p(X,Y,Z) = p(Z|X)p(Y|X)p(X)$. To enforce the IB principle, one needs to maximize the IB objective $I(Z;Y)-\beta I(Z;X)$, where $I(Z;Y)$ is the MI between the latent feature Z and its output Y, $I(Z;X)$ is the MI between the latent feature Z and its input X, and $\beta > 0$ controls the tradeoff between information compression and prediction accuracy.

Since $I(Z;Y) = H(Y)-H(Y|Z)$, where $H(Y)$ is the entropy of Y and is independent of the optimization procedure, optimizing the above IB objective is equivalent to minimizing the following IB Lagrangian:

$$\mathcal{L}_{\mathrm{IB}} = H(Y|Z) + \beta I(Z;X), \tag{1}$$

where $H(Y|Z) = \int p(\mathbf{y},\mathbf{z}) \log \frac{1}{p(\mathbf{y}|\mathbf{z})} d\mathbf{y} d\mathbf{z}$, for which introducing $q(\mathbf{y}|\mathbf{z})$ that appropriates $p(\mathbf{y}|\mathbf{z})$ gives a variational approximation of the intractable $p(\mathbf{y}|\mathbf{z})$. Specifically, since the KL divergence $D_{\mathrm{KL}}[p(\mathbf{y}|\mathbf{z}) \parallel q(\mathbf{y}|\mathbf{z})]$ is always non-negative, we have:

$$\int p(\mathbf{y}|\mathbf{z}) \log p(\mathbf{y}|\mathbf{z}) d\mathbf{y} \geq \int p(\mathbf{y}|\mathbf{z}) \log q(\mathbf{y}|\mathbf{z}) d\mathbf{y}, \tag{2}$$

which leads to

$$H(Y|Z) \leq -\int p(\mathbf{y},\mathbf{z}) \log q(\mathbf{y}|\mathbf{z}) d\mathbf{y} d\mathbf{z}, \tag{3}$$

where $p(\mathbf{y},\mathbf{z}) = \int p(\mathbf{x},\mathbf{y},\mathbf{z}) d\mathbf{x} = \int p(\mathbf{x}) p(\mathbf{y}|\mathbf{x}) p(\mathbf{z}|\mathbf{x}) d\mathbf{x}$ as denoted in Fig. 1(a). Thus, the upper bound of $H(Y|Z)$ is:

$$H(Y|Z) \leq -\int p(\mathbf{x},\mathbf{y}) p(\mathbf{z}|\mathbf{x}) \log q(\mathbf{y}|\mathbf{z}) d\mathbf{x} d\mathbf{y} d\mathbf{z}. \tag{4}$$

For the second term in Eq. (1), the upper bound is:

$$I(Z;X) \leq \int p(\mathbf{x}) p(\mathbf{z}|\mathbf{x}) \log \frac{p(\mathbf{z}|\mathbf{x})}{r(\mathbf{z})} d\mathbf{x} d\mathbf{z}, \tag{5}$$

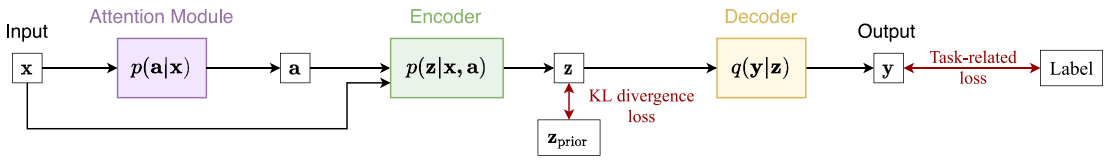


Fig. 2. Framework of the IB-inspired spatial attention mechanism for visual recognition. The input $\mathbf{x}$ is passed through an attention module to produce a continuous variational attention map $\mathbf{a}$. Then, $\mathbf{a}$ and $\mathbf{x}$ are encoded to a latent vector $\mathbf{z}$, and decoded to a prediction $\mathbf{y}$. See Section 3.2 for the deduction.

where $r(\mathbf{z})$ is a variational approximation to $p(\mathbf{z})$. By combining Eqs. (4) and (5), we have:

$$\begin{aligned}\mathcal{L}_{\mathrm{IB}} \leq & -\int p(\mathbf{x},\mathbf{y})p(\mathbf{z}|\mathbf{x})\log q(\mathbf{y}|\mathbf{z})d\mathbf{x}d\mathbf{y}d\mathbf{z} \\ & +\beta\int p(\mathbf{x})p(\mathbf{z}|\mathbf{x})\log\frac{p(\mathbf{z}|\mathbf{x})}{r(\mathbf{z})}d\mathbf{x}d\mathbf{z}.\end{aligned} \tag{6}$$

The above variational upper bound of the IB Lagrangian can be approximated using the empirical data distribution and the reparametrization trick (Kingma and Welling, 2014). See Alemi et al. (2017) for more details.

3.2. IB-inspired spatial attention mechanism

The main purpose of this paper is to develop an attention mechanism that shares the information constraint property of HVS by adapting the IB principle into our proposed attentive framework. In this way, the information is compressed when passing through the attention map, meanwhile, the most task-relevant part is preserved.

To this end, besides random variables X, Y, Z, we introduce a new random variable A that stands for an attention score map, and consider a new graphical model as shown in Fig. 1(b). The joint distribution of the four random variables is $p(X,Y,Z,A) = p(X)p(A|X)p(Y|X,A)p(Z|X,A)$. We aim to maximize the *Attentive IB* (*AIB*) objective: $I(Z;Y) - \beta I(Z;X,A)$, where $I(Z;X,A)$ is the MI between the latent feature Z and the joint distribution X, A. Inspired by the derivation of a representation learning framework from the IB principle as presented in Section 3.1, we obtain our IB-inspired spatial attention learning framework by deriving a new variational bound for the *AIB* objective.

Similarly, maximizing $I(Z;Y)$ can be achieved by minimizing $H(Y|Z)$, thus maximizing the *AIB* objective can be achieved by minimizing:

$$\mathcal{L}_{\mathrm{AIB}} = H(Y|Z) + \beta I(Z;X,A). \tag{7}$$

With Eq. (3), and by leveraging the fact that $p(\mathbf{y},\mathbf{z}) = \int p(\mathbf{x},\mathbf{y},\mathbf{z},\mathbf{a})d\mathbf{x}d\mathbf{a} = \int p(\mathbf{x})p(\mathbf{a}|\mathbf{x})p(\mathbf{y}|\mathbf{x},\mathbf{a})p(\mathbf{z}|\mathbf{x},\mathbf{a})d\mathbf{x}d\mathbf{a}$ (*c.f.* Fig. 1(b)), the new variational upper bound of $H(Y|Z)$ is calculated as follows:

$$H(Y|Z) \leq -\int p(\mathbf{x},\mathbf{y})p(\mathbf{a}|\mathbf{x})p(\mathbf{z}|\mathbf{x},\mathbf{a})\log q(\mathbf{y}|\mathbf{z})d\mathbf{x}d\mathbf{a}d\mathbf{y}d\mathbf{z}. \tag{8}$$

Next, we consider $I(Z;X,A)$, and obtain the following upper bound for it:

$$I(Z;X,A) \leq \int p(\mathbf{x})p(\mathbf{a}|\mathbf{x})p(\mathbf{z}|\mathbf{x},\mathbf{a})\log\frac{p(\mathbf{z}|\mathbf{x},\mathbf{a})}{r(\mathbf{z})}d\mathbf{x}d\mathbf{a}d\mathbf{z}, \tag{9}$$

where $r(\mathbf{z})$ is a variational approximation to $p(\mathbf{z})$, and we set $r(\mathbf{z})$ to be a spherical Gaussian $\mathcal{N}(\mathbf{z}|0,I)$ in our experiments. By combining Eq. (8) and (9), we obtain the upper bound for the $\mathcal{L}_{\mathrm{AIB}}$ defined in Eq. (7):

$$\begin{aligned}\mathcal{L}_{\mathrm{AIB}} \leq & -\int p(\mathbf{x},\mathbf{y})p(\mathbf{a}|\mathbf{x})p(\mathbf{z}|\mathbf{x},\mathbf{a})\log q(\mathbf{y}|\mathbf{z})d\mathbf{x}d\mathbf{a}d\mathbf{y}d\mathbf{z} \\ & +\beta\int p(\mathbf{x})p(\mathbf{a}|\mathbf{x})p(\mathbf{z}|\mathbf{x},\mathbf{a})\log\frac{p(\mathbf{z}|\mathbf{x},\mathbf{a})}{r(\mathbf{z})}d\mathbf{x}d\mathbf{a}d\mathbf{z}.\end{aligned} \tag{10}$$

Following Alemi et al. (2017), we approximate $p(\mathbf{x})$ and $p(\mathbf{x},\mathbf{y})$ with empirical distribution $p(\mathbf{x}) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{x}_n}(\mathbf{x})$ and $p(\mathbf{x},\mathbf{y}) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\mathbf{x}_n}(\mathbf{x})\delta_{\mathbf{y}_n}(\mathbf{y})$, respectively, where N is the number of training samples, $\mathbf{x}_n$ and $(\mathbf{x}_n,\mathbf{y}_n)$ are samples drawn from data distribution $p(\mathbf{x})$ and $p(\mathbf{x},\mathbf{y})$, respectively. The approximated upper bound of $\mathcal{L}_{\mathrm{AIB}}$ can be written as:

$$\begin{aligned}\widetilde{\mathcal{L}}_{\mathrm{AIB}} = & -\frac{1}{N}\sum_{n=1}^{N}\{\int p(\mathbf{a}|\mathbf{x}_n)p(\mathbf{z}|\mathbf{x}_n,\mathbf{a})\log q(\mathbf{y}_n|\mathbf{z})d\mathbf{a}d\mathbf{z} \\ & +\beta\int p(\mathbf{a}|\mathbf{x}_n)p(\mathbf{z}|\mathbf{x}_n,\mathbf{a})\log\frac{p(\mathbf{z}|\mathbf{x}_n,\mathbf{a})}{r(\mathbf{z})}d\mathbf{a}d\mathbf{z}\}.\end{aligned} \tag{11}$$

The above equation yields a neural framework as shown in Fig. 2 that is comprised of the following three modules:

- **Attention Module** $p(\mathbf{a}|\mathbf{x}_n)$ maps the input $\mathbf{x}_n$ to a continuous variational attention map $\mathbf{a}$. We use $\mathbf{a}$ to help compress the information of the input $\mathbf{x}$ that is passed to the latent code $\mathbf{z}$ and the output $\mathbf{y}$.
- **Encoder Module** $p(\mathbf{z}|\mathbf{x}_n,\mathbf{a})$ encodes the attention modulated input to a latent vector $\mathbf{z}$. Please refer to Section 3.3.1 for different schemes of modulating $\mathbf{x}$ using $\mathbf{a}$. We also employ different types of encoders to generate $\mathbf{z}$. See Section 3.3.2 for more details.
- **Decoder Module** $q(\mathbf{y}_n|\mathbf{z})$ predicts the task output $\mathbf{y}_n$ from the latent code $\mathbf{z}$.

We utilize Monte Carlo sampling to approximate the attention module and the encoder module. Suppose that we draw $L_{\mathbf{a}}$ times from $p(\mathbf{a}|\mathbf{x}_n)$ for $\mathbf{a}$ to get $\mathbf{a}^{(l_{\mathbf{a}})}$, and draw $L_{\mathbf{z}}$ times from $p(\mathbf{z}|\mathbf{x}_n,\mathbf{a}^{(l_{\mathbf{a}})})$ for $\mathbf{z}$ to get $\mathbf{z}^{(l_{\mathbf{a}},l_{\mathbf{z}})}$. Then we can optimize our framework by minimizing the following loss function induced from Eq. (11) by applying the above Monte Carlo sampling:

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\{\frac{1}{L_{\mathbf{a}}L_{\mathbf{z}}}\sum_{l_{\mathbf{a}}=1}^{L_{\mathbf{a}}}\sum_{l_{\mathbf{z}}=1}^{L_{\mathbf{z}}}\mathcal{L}_{\mathrm{task}} + \beta\frac{1}{L_{\mathbf{a}}}\sum_{l_{\mathbf{a}}=1}^{L_{\mathbf{a}}}\mathcal{L}_{D_{\mathrm{KL}}}\}, \tag{12}$$

in which,

$$\mathcal{L}_{\mathrm{task}} = \log q(\mathbf{y}_n|\mathbf{z}^{(l_{\mathbf{a}},l_{\mathbf{z}})}) \tag{13}$$

is a task-related loss. For different tasks, we apply different task-related losses on the output $\mathbf{y}_n$ (*c.f.* Section 3.4.2). And,

$$\mathcal{L}_{D_{\mathrm{KL}}} = D_{\mathrm{KL}}[p(\mathbf{z}|\mathbf{x}_n,\mathbf{a}^{(l_{\mathbf{a}})})\|\,r(\mathbf{z})] \tag{14}$$

is the KL divergence loss which enforces the similarity between the prior distribution $r(\mathbf{z})$ and the distribution of $\mathbf{z}$ generated by the encoder module (*c.f.* Section 3.4.1).

3.2.1. Attention score quantization

Before looking into the above three modules, we describe an attention score quantization strategy. We define the continuous attention space as $\mathcal{A} \in \mathbb{R}^{W\times H}$, and the quantized attention space as $\mathcal{A}_q \in \mathbb{R}^{W\times H}$, where W, H are the width and height of the attention map, respectively, same as those of the input $\mathbf{x}$. As shown in Fig. 3, the input $\mathbf{x}$ is passed through an attention module to produce a continuous variational attention map $\mathbf{a}$, which is mapped to a discrete attention map $\mathbf{a}_q$ through a nearest neighbor look-up among a set of learnable anchor values $\{v_i \in \mathbb{R}\}_{i=1}^{Q}$:

$$\mathbf{a}_q^{(w,h)} = v_k,\ k = \arg\min_j \|\mathbf{a}^{(w,h)} - v_j\|_2, \tag{15}$$

where $w = 1\cdots W, h = 1\cdots H$ are spatial indices. In this way, each score $\mathbf{a}^{(w,h)}$ in the continuous attention map is mapped to the 1-of-Q anchor value. The attention quantization makes the following possible: instead of the continuous attention $\mathbf{a}$, the quantized attention map $\mathbf{a}_q$

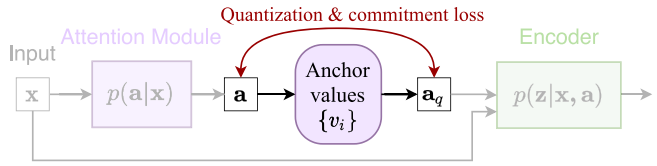


Fig. 3. Attention score quantization module. To further restrict the information bypassed by the attention map, the continuous variational attention map $\mathbf{a}$ is quantized to a discrete attention map $\mathbf{a}_q$ using a set of learnable anchor values $\{v_i\}$. Then, $\mathbf{a}_q$ and $\mathbf{x}$ instead of $\mathbf{a}$ and $\mathbf{x}$ are encoded to a latent vector $\mathbf{z}$. See Section 3.2.1 for more details.

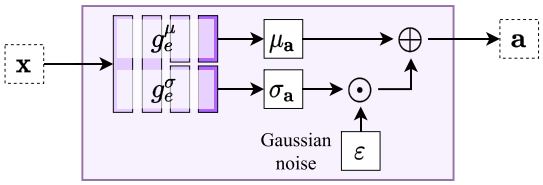


Fig. 4. The attention module is designed to output attention maps that follows $\mathcal{N}(\mu_{\mathbf{a}}, \sigma_{\mathbf{a}})$. See Section 3.3.1 for more details.

and the input $\mathbf{x}$ are encoded into a latent representation $\mathbf{z} \in \mathbb{R}^K$ in a K dimensional latent space. In this way, we can further compress the information filtered by the attention maps.

As the arg min operation in Eq. (15) is not differentiable, we resort to the straight-through estimator (Bengio et al., 2013) and approximate the gradient of $\mathbf{a}_q$ using the gradient of $\mathbf{a}$. Though simple, this estimator works well for the experiments in this paper. Specifically, in the forward process the quantized attention map $\mathbf{a}_q$ is passed to the encoder, and during the backward computation, the gradient of $\mathbf{a}$ is passed to the attention module unaltered. Such a gradient approximation makes sense because $\mathbf{a}_q$ and $\mathbf{a}$ share the same $W \times H$ dimensional space, and the gradient of $\mathbf{a}$ can provide useful information on how the attention module could change its output to minimize the loss function defined in Eq. (12).

We add two extra loss terms to learn the anchor values $\{v_i\}$ and optimize $\mathbf{a}_q$, namely a quantization objective $\mathcal{L}_{\text{vq}}$ and a commitment term $\mathcal{L}_{\text{cmt}}$:

$$\mathcal{L}_{\text{vq}} = \|\text{sg}[\mathbf{a}^{(l_{\mathbf{a}})}] - \mathbf{a}_q^{(l_{\mathbf{a}})}\|_2^2, \tag{16}$$

$$\mathcal{L}_{\text{cmt}} = \|\mathbf{a}^{(l_{\mathbf{a}})} - \text{sg}[\mathbf{a}_q^{(l_{\mathbf{a}})}]\|_2^2, \tag{17}$$

where $\text{sg}[\cdot]$ is the stopgradient operator (Van Den Oord et al., 2017), and $\mathbf{a}_q^{(l_{\mathbf{a}})}$ is the quantized version of $\mathbf{a}^{(l_{\mathbf{a}})}$. Specifically, $\mathcal{L}_{\text{vq}}$ updates the anchor values to move towards the attention map $\mathbf{a}$, and $\mathcal{L}_{\text{cmt}}$ forces the attention module to commit to the anchor values.

3.3. Module design choices

In the above, we have introduced our spatial attention learning framework derived from the *AIB* objective, as well as the network modules within the framework. In the following, we introduce the different realization methods of the network modules, which are closely related to the applications.

3.3.1. Attentive feature modulation strategies

The input $\mathbf{x}$ is modulated with the learned attention map $\mathbf{a}$ to yield the latent embedding $\mathbf{z}$. We suppose to use the attention module of the form $p(\mathbf{a}|\mathbf{x}) = \mathcal{N}(\mathbf{a}|\mu_{\mathbf{a}}, \text{diag}(\sigma_{\mathbf{a}}^2))$, where $\mu_{\mathbf{a}} = g_e^{\mu}(\mathbf{x})$, $\sigma_{\mathbf{a}} = g_e^{\sigma}(\mathbf{x})$, and g_e^{μ}, g_e^{σ} are network modules. To enable back-propagation, we use the reparametrization trick (Kingma and Welling, 2014), and obtain $\mathbf{a}^{(l_{\mathbf{a}})}$ by drawing an $\varepsilon^{(l_{\mathbf{a}})}$ from $\mathcal{N}(0, I)$ and calculate $\mathbf{a}^{(l_{\mathbf{a}})} = \mu_{\mathbf{a}} + \sigma_{\mathbf{a}} \cdot \varepsilon^{(l_{\mathbf{a}})}$. An illustration of the attention module is shown in Fig. 4. In the following, we discuss different strategies to modulate the feature using the spatial attention map $\mathbf{a}$. The effectiveness of different modulation strategies are analyzed in Section 4.1.4.

Scaling ($\odot$). A general way to modulate a feature with an attention map, is to perform pixel-wise multiplication to weigh each spatial location differently, *i.e.*, $\mathbf{x}_{\text{att}}^c = \mathbf{x}^c \odot \mathbf{a}$, where $\mathbf{x}^c$ and $\mathbf{x}_{\text{att}}^c$ indicate the cth slices of the input $\mathbf{x}$ and the modulated feature $\mathbf{x}_{\text{att}}$, respectively, and $\odot$ denotes Hadamard product.

Scaling with residual ($\odot\oplus$). Besides directly scaling the feature, an extra residual connection (He et al., 2016) can be used to ease the learning of attention. The attention modulated feature is calculated as $\mathbf{x}_{\text{att}}^c = \mathbf{x}^c \oplus (\mathbf{x}^c \odot \mathbf{a})$, where $\oplus$ represents pixel-wise addition, and c is the channel index.

Apart from the scaling mentioned above, we also consider modulating the input by directly biasing it with the attention map, *i.e.*, $\mathbf{x}_{\text{att}} = \mathbf{x} \oplus U(\mathbf{a})$, where U is an operation that maps $\mathbf{a}$ to a feature that have the same channels as $\mathbf{x}$, which is modeled using a 3×3 convolution layer with Sigmoid activation.

Concatenation (*cat*). The fourth way for feature modulation is to obtain the modulated feature by concatenation, which gives $\mathbf{x}_{\text{att}} = cat(\mathbf{x}, U(\mathbf{a}))$, where *cat* represents concatenation, and U is a channel upsampling module same as above.

3.3.2. Different types of encoders

We consider three types of encoders, which are shown in Fig. 5. The study of different encoders is presented in Section 4.1.5.

Gaussian encoder. This type of encoder (Fig. 5(a)) yields latent embedding $\mathbf{z}$ with a tractable density, *i.e.*, $\mathbf{z} \sim \mathcal{N}(\mu_n, \text{diag}(\sigma_n^2))$, which follows a Gaussian distribution with mean $\mu_n = f_e^{\mu}(\mathbf{x}_n, \mathbf{a}^{(l_{\mathbf{a}})})$ and standard derivation $\sigma_n = f_e^{\sigma}(\mathbf{x}_n, \mathbf{a}^{(l_{\mathbf{a}})})$ both conditioned on the inputs of the encoder, where f_e^{μ} and f_e^{σ} are network modules. We use the reparametrization trick (Kingma and Welling, 2014), and obtain $\mathbf{z}^{(l_{\mathbf{a}}, l_{\mathbf{z}})}$ by first drawing an $\epsilon^{(l_{\mathbf{z}})}$ from $\mathcal{N}(0, I)$ and then calculating $\mathbf{z}^{(l_{\mathbf{a}}, l_{\mathbf{z}})} = \mu_n + \sigma_n \cdot \epsilon^{(l_{\mathbf{z}})}$. Since the density of $\mathbf{z}$ is tractable, the KL divergence loss defined in Eq. (14) can be estimated either in a closed-form or with adversarial learning, as will be detailed in Section 3.4.1.

Additive noise encoder. This type of encoder (Fig. 5(b)) introduces the stochasticity to its output by adding a noise to the input $\mathbf{x}$. In this way, $\mathbf{z}^{(l_{\mathbf{a}}, l_{\mathbf{z}})} = Enc(\mathbf{x}^{(l_{\mathbf{z}})}, \mathbf{a}^{(l_{\mathbf{a}})})$, where $\mathbf{x}^{(l_{\mathbf{z}})} = \mathbf{x} + \sigma \cdot \epsilon^{(l_{\mathbf{z}})}$, $\epsilon^{(l_{\mathbf{z}})}$ is drawn from a normal distribution, $\sigma \in \mathbb{R}^+$ is the standard deviation of the noise, and $\mathbf{a}^{(l_{\mathbf{a}})}$ is calculated from $\mathbf{x}$ using the attention module. For this encoder, only the adversarial learning approach can be used to estimate the KL divergence loss defined in Eq. (14).

Propagated noise encoder. This type of encoder (Fig. 5(c)) propagates the noise ϵ across the input $\mathbf{x}$ through concatenation, which gives $\mathbf{x}^{(l_{\mathbf{z}})} = cat(\mathbf{x}, \sigma \cdot \epsilon^{(l_{\mathbf{z}})})$, where σ and $\epsilon^{(l_{\mathbf{z}})}$ are the same as above. We then calculate $\mathbf{z} = Enc(\mathbf{x}^{(l_{\mathbf{z}})}, \mathbf{a}^{(l_{\mathbf{a}})})$, where $\mathbf{a}^{(l_{\mathbf{a}})}$ is the attention map computed from $\mathbf{x}$. For this encoder, also only the adversarial learning approach can be used to estimate the KL divergence loss defined in Eq. (14).

3.4. Losses

The complete model parameters include the parameters of the attention module, encoder, decoder, and the anchor values $\{v_i \in \mathbb{R}\}_{i=1}^{Q}$. The overall loss function is defined as in Eq. (18), which is composed of four terms.

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\frac{1}{L_{\mathbf{a}}}\sum_{l_{\mathbf{a}}=1}^{L_{\mathbf{a}}}\Big\{\Big(\frac{1}{L_{\mathbf{z}}}\sum_{l_{\mathbf{z}}=1}^{L_{\mathbf{z}}}\mathcal{L}_{\text{task}}\Big) + \beta\mathcal{L}_{D_{\text{KL}}} + \lambda_{\text{vq}}\mathcal{L}_{\text{vq}} + \lambda_{\text{cmt}}\mathcal{L}_{\text{cmt}}\Big\}. \tag{18}$$

We set $\beta = 0.01$, $\lambda_{\text{vq}} = 0.4$, and $\lambda_{\text{cmt}} = 0.1$ empirically.

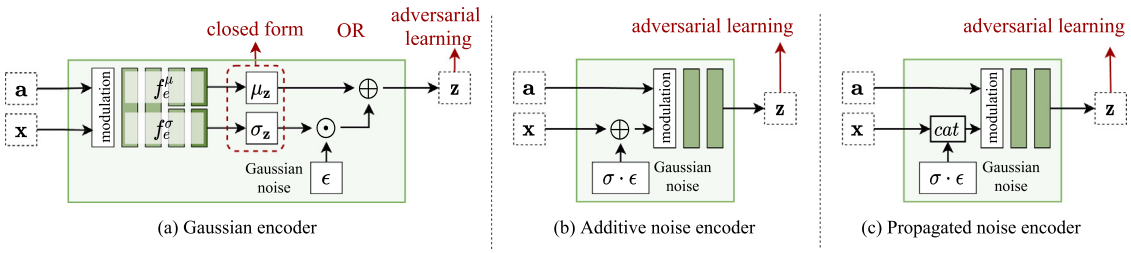


Fig. 5. Types of encoders. We consider three types of encoders (Section 3.3.2), namely Gaussian encoder, additive noise encoder, and additive noise encoder. Here, $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, and σ is the standard derivation of the noise ϵ. For the Gaussian encoder, the KL divergence loss ($\mathcal{L}_{D_{\mathrm{KL}}}$) can be calculated in closed form or approximated using adversarial learning. For additive or propagated noise encoders, the KL divergence loss is minimized with adversarial learning. We also explore various attentive feature modulation strategies (Section 3.3.1).

3.4.1. Evaluating KL divergence loss

The $\mathcal{L}_{D_{\mathrm{KL}}}$ defined in Eq. (14) measures the distance between the distribution of latent encoding $\mathbf{z}$ and the prior $r(\mathbf{z})$.

Closed-form Evaluation. When using Gaussian encoder, the KL divergence between the latent encoding $\mathbf{z} \sim \mathcal{N}(\mu_n, \mathrm{diag}(\sigma_n^2))$ and the prior $\mathbf{z} \sim \mathcal{N}(0, I)$ is:

$$\mathcal{L}_{D_{\mathrm{KL}}} = -\frac{1}{2K} \sum_{i=1}^{K} \left(1 + \log \sigma_{n,i}^2 - \mu_{n,i}^2 - \sigma_{n,i}^2\right), \tag{19}$$

where i is the element index, K is the dimension of $\mathbf{z}$, $\mu_n = f_e^\mu(\mathbf{x}_n, \mathbf{a}^{(l_\mathbf{a})})$ and $\sigma_n = f_e^\sigma(\mathbf{x}_n, \mathbf{a}^{(l_\mathbf{a})})$ are the mean and standard deviation vectors, respectively, and f_e^μ, f_e^σ form the Gaussian encoder.

Adversarial Learning based Evaluation. The above closed-form of the KL divergence requires the learned latent encoding to have a tractable density, which inevitably restricts the learning ability of the framework. Inspired by Makhzani et al. (2016), Hjelm et al. (2019), we train a discriminator D to estimate the KL divergence between the learned distribution $p(\mathbf{z}|\mathbf{x}, \mathbf{a})$ and the prior $r(\mathbf{z})$. This enables us to be also able to utilize the last two types of encoders introduced in Section 3.3.2. Meanwhile, the Gaussian encoder can also use this type of KL divergence evaluation method. Specifically, the training objective is:

$$\begin{aligned}\mathcal{L}_{D_{\mathrm{KL}}}(Enc, D) = &\mathbb{E}_{\mathbf{z}\sim r(\mathbf{z})}[\log D(\mathbf{z})] \\ &+ \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log(1 - D(Enc(\mathbf{x}, \mathbf{a})))],\end{aligned} \tag{20}$$

where $\mathbb{E}[\cdot]$ is the expectation operator. The input to the discriminator D is either the sample $\mathbf{z}$ drawn from the latent prior $r(\mathbf{z})$, or the latent vector yield by the encoder Enc. During training, we minimize Eq. (20) w.r.t. the parameters of the encoder network Enc, while maximize it w.r.t. the parameters of the discriminator D. This process is implemented by iterating the following two steps: (1) *Training the discriminator.* Training D is achieved by maximizing $\mathcal{L}_{D_{\mathrm{KL}}}(Enc, D)$ with Enc fixed. In this way, D is trained to output a high score for the samples drawn from the latent prior, and a low score for the latent vectors generated by the encoder. (2) *Training the encoder.* In this step, D is fixed, and $\mathcal{L}_{D_{\mathrm{KL}}}(Enc) = \mathbb{E}_{\mathbf{x}\sim p(\mathbf{x})}[\log(1 - D(Enc(\mathbf{x}, \mathbf{a})))]$ is minimized, which is part of $\mathcal{L}_{D_{\mathrm{KL}}}(Enc, D)$ as defined in Eq. (20). The encoder network is expected to achieve a higher score when evaluated using D. Through the above training process, D learns to distinguish between the two distributions, and Enc is trained to match the latent prior $r(\mathbf{z})$ implicitly.

3.4.2. Task-related losses

The $\mathcal{L}_{\mathrm{task}} = \log q(\mathbf{y}_n|\mathbf{z}^{(l_\mathbf{a}, l_\mathbf{z})})$ in Eq. (18) is closely related to the task performance. In the following, we introduce the concrete forms of this term in two types of representative tasks, namely visual recognition and dense prediction.

Visual Recognition Tasks. In visual recognition tasks, $\mathcal{L}_{\mathrm{task}}$ is typically modeled as a cross-entropy loss (Belghazi et al., 2018):

$$\mathcal{L}_{\mathrm{task}} \equiv \mathcal{L}_{\mathrm{cls}} = -\frac{1}{C} \sum_{c=1}^{C} \mathscr{e}_n^c \log(\mathbf{y}_n^c), \tag{21}$$

where $\mathbf{y}_n^c$ is the prediction score for the cth category, $\mathscr{e}_n^c \in \{0, 1\}$ denotes whether the image can be categorized as class c, and C is the number of classes.

Semantic Segmentation. Different from visual recognition tasks that predict a label for an input, semantic segmentation assigns a label to every pixel in an image. The task loss $\mathcal{L}_{\mathrm{task}}$ for semantic segmentation is defined as:

$$\mathcal{L}_{\mathrm{task}} \equiv \mathcal{L}_{\mathrm{seg}} = -\frac{1}{W \times H \times C} \sum_{i=1}^{W \times H \times C} \mathbf{Y}_n^i \log(\mathbf{y}_n^i), \tag{22}$$

where $\mathbf{y}_n$ is the predicted mask, $\mathbf{Y}_n \in \mathbb{R}^{W \times H \times C}$ is the ground-truth semantic label for the input data $\mathbf{x}_n$ with resolution $W \times H$, and C is the number of semantic categories.

Object Detection. The task loss $\mathcal{L}_{\mathrm{task}}$ for object detection consists of two primary components:

$$\mathcal{L}_{\mathrm{task}} \equiv \mathcal{L}_{\mathrm{det}} = \mathcal{L}_{\mathrm{cls}} + \lambda_{\mathrm{reg}} \mathcal{L}_{\mathrm{reg}}. \tag{23}$$

Here, $\mathcal{L}_{\mathrm{cls}}$ is the classification loss, typically cross-entropy loss, which measures how well the predicted class probabilities match the ground truth labels. The second term $\mathcal{L}_{\mathrm{reg}}$ is the regression loss, typically Smooth L1 loss, which measures how well the predicted bounding box coordinates match the ground truth box coordinates. The parameter λ_{reg} weights the importance of the regression loss relative to the classification loss, which is often set to 1.

4. Experiments

4.1. Ablation studies

In this section, we conduct ablation studies on hyperparameters and evaluate the robustness of the proposed framework w.r.t. different realizations. Note that our model can be applied to various kinds of downstream tasks. Here, we only perform the ablation study on the task of image classification. We first introduce the experiment setup, and then show the results of the ablation studies.

4.1.1. Experiment setup

Datasets. We use two datasets for the ablation studies, namely CIFAR-10 (Krizhevsky et al., 2009) and CIFAR-100 (Krizhevsky et al., 2009). CIFAR-10 contains 60,000 32×32 natural images of 10 classes, which are split into 50,000 training and 10,000 test images. CIFAR-100 is similar to CIFAR-10, except that it has 100 classes. We use original input images after data augmentation (random flipping and cropping with a padding of 4 pixels).

Network configurations. Our IB-inspired attention learning framework consists of an encoder, an attention module, and a decoder. In practice, instead of directly feeding the input image into the encoder and attention modules, we transform it into the feature space by a feature extractor.

We build our framework by extending standard backbone networks. Specifically, we consider two backbone networks, namely VGG (Simonyan

Table 1
Network configuration of visual recognition tasks on CIFAR datasets. Here, `block` denotes the basic building block of the backbone, and K is the dimension of the latent space. See Section 4.1.1 for more details about the experiments.

Backbone	Feature extractor	Attention module	Encoder	Decoder
VGG16	`block1-3`	g_e^μ: $Conv(3\times3,1)$+$Sigmoid$ g_e^σ: $Conv(3\times3,1)$	Gaussian: `block4-5`+$Conv(1\times1,512)$+$FC(512,2K)$ Additive: `block4-5`+$Conv(1\times1,512)$+$FC(512,K)$ Propagated: `block4-5`+$Conv(1\times1,512)$+$FC(512,K)$	$FC(K,K)$+$ReLU$+$FC(K,C)$
WRN28-10	`block1-2`	g_e^μ: `block3`+$Conv(3\times3,1)$+$Sigmoid$ g_e^σ: $Conv(3\times3,1)$	Gaussian: `block3`+$GlobalPooling$+$FC(640,2K)$	$ReLU$+$FC(K,C)$

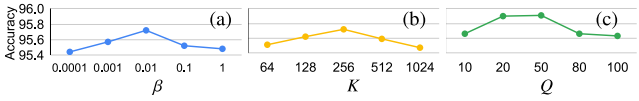


Fig. 6. Analysis of hyperparameters on CIFAR-10 with VGG backbone. See Section 4.1.2 for more details.

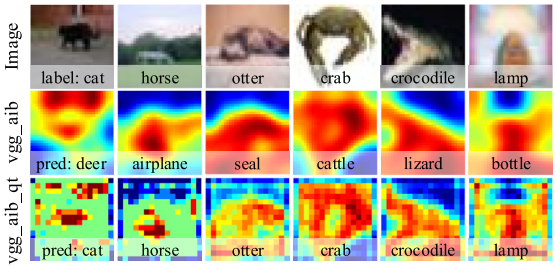


Fig. 7. Effect of attention score quantization. See Section 4.1.2 for details.

and Zisserman, 2015) and WRN (Zagoruyko and Komodakis, 2016). As shown in Table 1, when using VGG as the backbone, the feature extractor is built from the first three convolutional blocks of VGG16 with the max-pooling layers omitted. For the attention module, we simply use two convolutional network modules for learning the mean and standard deviation of the attention map, respectively. For the encoder, it first uses the block4-5 of VGG16 and then a 1×1 convolutional layer, followed by an extra fully-connected (FC) layer. For the Gaussian encoder, the FC layer maps the flattened attention-modulated feature to a $2K$-dim vector, where the first K-dim is μ, and the remaining K-dim is σ (after a softplus activation), resulting latent encoding $\mathbf{z}=\mu+\sigma\cdot\epsilon\in\mathbb{R}^K$ where K is the dimension of the latent encoding. For the additive and the propagated noise encoders, the FC layer directly outputs the K-dim vector $\mathbf{z}$. Finally, the decoder, which consists of two FC layers, maps the $\mathbf{z}$ to the prediction $\mathbf{y}\in\mathbb{R}^C$, where C is the number of classes.

When using WRN as the backbone, the first two residual blocks of WRN28-10 are used as the feature extractor. The attention module is similar to that of the VGG-backboned framework, except that the network module for learning the mean value contains the third `block` from WRN. For the WRN backbone, we only experiment with the Gaussian encoder, which is composed of the duplicated third `block` of WRN, followed by a global pooling layer, and an FC layer. The decoder consists of a single FC layer.

In the following, we use "VGG-aib" to denote the above VGG-based model, and "WRN-aib" to denote the WRN-based model. We also design "VGG-aib-qt" and "WRN-aib-qt" which use the proposed attention score quantization strategy, and the number of anchor values Q is 20 and 50, respectively. The anchor values $\{v_i\}$ are initialized by evenly-spaced points in $[0,1]$, and trained end-to-end in the framework.

Training and inference. Models on CIFARs are trained from scratch following the previous practice. We use an SGD optimizer with weight decay 5×10^{-4} and momentum 0.9, and train for 200 epochs in total. The training batch size is 128, and the testing batch size is 100. For "VGG-aib(-qt)", the initial learning rate is 0.1, which is scaled by 0.5 every 25 epochs. For "WRN-aib(-qt)", the initial learning rate is 0.1, and is multiplied by 0.3 at the 60,120 and 180-th epoch. Without loss of generality, we draw $L_{\mathbf{a}}=1$ samples for $\mathbf{a}$ and $L_{\mathbf{z}}=1$ samples for $\mathbf{z}$ for the loss calculated in Eq. (12) and (18) during training, and set $L_{\mathbf{a}}=L_{\mathbf{z}}=4$ during inference.

4.1.2. Ablations on hyperparameters

Firstly, we conduct ablation studies on hyperparameters. When testing different hyperparameters, we always use VGG backbone, the Gaussian encoder, the "scaling with residual" ($\odot\oplus$) feature modulation, and the closed-form KL divergence loss.

Information tradeoff factor β controls the amount of information that bypasses the bottleneck. To measure the influence of β, we train "VGG-aib" model on CIFAR-10 with different β values using loss function Eq. (12). The classification accuracy of the resulted models are plotted in Fig. 6(a). As can be observed, the accuracy is largest when $\beta=0.01$.

Latent vector dimension K. We experiment on $K=64,128,256,512,1024$. As shown in Fig. 6(b), "VGG-aib" achieves the best performance when $K=256$.

Attention score quantization number Q determines the granularity of the quantization which further affects the information bypassed by the spatial attention map. Fig. 6(c) presents the classification accuracy of "VGG-aib-qt" with $\beta=0.01, K=256$ when varying number of anchor values Q, showing that Q between 20 and 50 gives better performance. We use $Q=20$ in default. Furthermore, exemplary attention maps of cases that are wrongly classified by "VGG-aib" but are correctly predicted by "VGG-aib-qt" are listed in Fig. 7. As can be seen, the attention quantization module can further help focus on more concentrated regions with important information, thus improving the accuracy.

4.1.3. Ablations on feature extraction backbones

As stated above, we consider two backbones, *i.e.*, VGG (Simonyan and Zisserman, 2015) and WRN (Zagoruyko and Komodakis, 2016) that were designed for visual recognition. When testing different backbones, we always use the Gaussian encoder, the "scaling with residual" ($\odot\oplus$) feature modulation, and the closed-form KL loss.

By comparing Line 2 and Line 3 in Table 2 with the *vanilla* models in Line 1 and Line 0, respectively, we conclude that our proposed attention mechanism can improve different backbones. Specifically, we achieve a relative error decrease of 44.92%/47.23% and 29.59%/31.84% for "VGG-aib/VGG-aib-qt" over *vanilla* VGG on CIFAR10 and CIFAR100, respectively, and a relative error decrease of 10.00%/14.25% and 7.43%/8.36% for "WRN-aib/WRN-aib-qt" over *vanilla* WRN on CIFAR10 and CIFAR100, respectively. As can be seen, WRN is a better backbone than VGG.

4.1.4. Ablations on attentive feature modulation

We explore four strategies for modulating the feature using the learned spatial attention map, namely, "scaling" ($\odot$), "scaling with residual" ($\odot\oplus$), "biasing" ($\oplus$), and "concatenation" (cat). More details can be found in Section 3.3.1. To evaluate different feature modulation methods, we always adapt the VGG backbone, Gaussian encoder, and closed-form KL loss.

Table 2
Robustness of the IB-inspired spatial attention learning framework on various framework configurations. Here presents the error rates (%) for image classification on the CIFAR-10 and CIFAR-100 datasets. Lower error rates indicate better classification performance. See Section 4.1 for more details.

No.	Model	Backbone	Att. Modulation	Enc. Types	KL Loss Eva.	CIFAR-10	CIFAR-100
0	*vanilla* VGG	–	–	–	–	7.77	30.62
1	*vanilla* WRN	–	–	–	–	4.00	19.25
2	WRN-aib WRN-aib-qt	WRN	$\odot\oplus$	Gaussian	Closed form (Eq. (19))	3.60 3.43	17.82 17.64
3	VGG-aib VGG-aib-qt	VGG	$\odot\oplus$	Gaussian	Closed form (Eq. (19))	4.28 4.10	21.56 20.87
4	VGG-aib VGG-aib-qt	VGG	$\odot$	Gaussian	Closed form (Eq. (19))	4.40 4.33	21.74 21.58
5	VGG-aib VGG-aib-qt	VGG	$\oplus$	Gaussian	Closed form (Eq. (19))	4.34 4.25	21.64 21.38
6	VGG-aib VGG-aib-qt	VGG	*cat*	Gaussian	Closed form (Eq. (19))	4.39 4.31	21.77 21.55
7	VGG-aib VGG-aib-qt	VGG	$\odot\oplus$	Gaussian	Adv. Learning (Eq. (20))	4.20 4.09	21.60 21.17
8	VGG-aib VGG-aib-qt	VGG	$\odot\oplus$	Additive	Adv. Learning (Eq. (20))	4.29 4.08	21.69 21.46
9	VGG-aib VGG-aib-qt	VGG	$\odot\oplus$	Propagated	Adv. Learning (Eq. (20))	4.28 4.15	21.62 21.50

Table 3
Running time analysis with various task settings. See Section 4.1.7 for details.

Input Size	#Class	FLOPs (G)				Inference time per frame (ms)			
		VGG-aib	VGG-aib-qt	WRN-aib	WRN-aib-qt	VGG-aib	VGG-aib-qt	WRN-aib	WRN-aib-qt
32×32	10	3.79	3.79	39.14	39.14	3.43	3.68	8.53	8.55
32×32	100	3.79	3.79	39.20	39.20	3.51	3.70	8.60	8.84
224×224	200	16.75	16.75	50.97	50.97	3.58	3.81	10.47	10.66

The performance of different attentive feature modulation strategies is shown in Table 2, Line 3–6. As can be seen, all the strategies achieve comparable recognition errors, while "scaling with residual" ($\odot\oplus$) performs better than all others, especially for "VGG-aib-qt" on CIFAR-100. That is why we mostly use the "scaling with residual" feature modulation strategy in all other experiments unless otherwise specified.

4.1.5. Ablations on encoder types

We consider three types of encoders, namely Gaussian, additive noise, and propagated noise encoders, following Alemi et al. (2017), Belghazi et al. (2018). More detailed introductions of the encoders are shown in Section 3.3.2. We experiment with different types of encoders but with the same VGG backbone, "scaling with residual" attentive feature modulation, and adversarial learning for evaluating the KL loss considering that the outputs of the two noise encoders are intractable.

Table 2, Lines 7–9 present the classification errors of our proposed attention learning framework with three types of encoders. The three types of encoders achieve on-par performance with each other, indicating the robustness of our framework towards various stochastic latent encoding learning. By default, we use Gaussian encoder.

4.1.6. Ablations on KL divergence loss evaluation

In previous works (Alemi et al., 2017; Lai et al., 2021), the mutual information estimation is typically calculated in closed form, which requires the density of the encoded features to be tractable. To facilitate a more flexible design, we utilize adversarial learning to estimate the loss term $\mathcal{L}_{D_{\mathrm{KL}}}$. The closed-form and the adversarial learning KL loss evaluation methods are introduced in Section 3.4.1.

When accessing the two kinds of loss evaluation strategies, we use the VGG backbone, Gaussian encoder, and "scaling with residual" attentive feature modulation. The discriminator D for adversarial training is built as: $FC(K, K) \rightarrow LeakyReLU \rightarrow FC(K, K) \rightarrow LeakyReLU \rightarrow FC(K, 1) \rightarrow Sigmoid$, and the dropout rate of each FC layer is 0.2. We can conclude from Line 3 and 7, Table 2 that our IB-inspired attention learning framework is robust to the ways of evaluating the KL loss. In default, we use the closed-form KL evaluation method as the default encoder type is Gaussian in all other experiments.

4.1.7. Running time analysis

We conduct experiments on a single NVIDIA Titan Xp GPU, and report the floating-point operations per second (FLOPs) and inference time of our method in Table 3. As can be observed, models employing a WRN backbone incur higher computational costs compared to those with a VGG backbone, showing increased FLOPs and extended inference time. Within models utilizing the same backbone architecture, FLOPs increase with the input resolution because more convolution operations are needed to cover the input given the 3×3 kernels. The attention score quantization module only slightly increases the inference time, and exerts a negligible impact on the FLOPs.

4.2. Comparison with state-of-the-arts

Besides the image classification task, we apply our IB-inspired spatial attention method to many other visual recognition tasks, including fine-grained recognition and cross-dataset classification. We also apply it to two other computer vision tasks, namely, semantic segmentation and object detection. In the following, we compare our method with the state-of-the-arts on these tasks. In these experiments, our attention method achieves improved performance over the state-of-the-art methods, which demonstrates the effectiveness of our method.

For all the experiments in this section, we use the framework configurations listed in Line 2 and Line 3, Table 2, *i.e.*, we fix the encoder type to be Gaussian, use "scaling with residual" attentive feature modulation strategy, and evaluate KL divergence loss in closed form. The detailed structures for different framework components are summarized in Table 1 and 4. The task losses, backbones, and network configurations for each task will be introduced in the following.

Table 4
Network configuration of visual recognition tasks on CUB. Here, `block` denotes the basic building block of the backbone, and K is the dimension of the latent space. See Section 4.2.2 for more details about the experiments.

Backbone	Feature Extractor	Attention Module	Encoder	Decoder
VGG16	`block1-5`	g_e^μ: $Conv(3\times3,1)$+$Sigmoid$ g_e^σ: $Conv(3\times3,1)$	Gaussian: $AvgPooling$+$FC(512,2K)$	$FC(K,K)$+$ReLU$+$FC(K,C)$
WRN50-2	`block1-3`	g_e^μ: `block4`+$Conv(3\times3,1)$+$Sigmoid$ g_e^σ: $Conv(3\times3,1)$	Gaussian: `block4`+$AvgPooling$+$FC(2048,2K)$	$FC(K,K)$+$ReLU$+$FC(K,C)$

Table 5
Top-1 error for image classification (Section 4.2.1), fine-grained recognition (Section 4.2.2), and cross-dataset classification (Section 4.2.3). Here, "aib" denotes "attentive information bottleneck" and "qt" denotes "quantization". * denotes re-implementation or re-training. Other values are from the original paper. Best values of different backbones are in **bold**.

Model	Image Class.		Fine-grained Recog.		Cross-domain Class.	
	CIFAR-10	CIFAR-100	CUB-200–2011	SVHN	STL10-train	STL10-test
– Architectures without attention –						
VGG	7.77	30.62	34.64	4.27	54.66	55.09
VGG-GAP	9.87	31.77	29.50	5.84	56.76	57.24
VGG-PAN	6.29	24.35	31.46	8.02	52.50	52.79
VGG-DVIB	4.64*	22.88*	**23.94***	3.28*	**51.40***	**51.60***
WRN	**4.00**	**19.25**	26.50	–	–	–
– Architectures with attention –						
VGG-att2	5.23	23.19	26.80	3.74	51.24	51.71
VGG-att3	6.34	22.97	26.95	3.52	51.58	51.68
VGG-aib (ours)	4.28	21.56	23.73	3.24	50.64	51.24
VGG-aib-qt (ours)	**4.10**	**20.87**	**21.83**	**3.07**	**50.44**	**51.16**
WRN-ABN	3.92*	18.12	–	2.88*	50.90*	51.24*
WRN-aib (ours)	3.60	17.82	17.26	2.76	**50.08**	50.84
WRN-aib-qt (ours)	**3.43**	**17.64**	**16.88**	**2.69**	50.34	**50.49**

4.2.1. Image classification

The experiment setup for the image classification task on CIFAR datasets is the same as presented in Section 4.1.1.

Results. As shown in Table 5, the proposed attention mechanism achieves noticeable performance improvement over standard architectures and existing attention mechanisms such as GAP (Zhou et al., 2016), PAN (Seo et al., 2018), VGG-att[1] (Jetley et al., 2018) and ABN.[2] (Fukui et al., 2019) To be specific, "VGG-aib" achieves 3.49% and 9.06% decrease of errors over the baseline VGG model on CIFAR-10 and CIFAR-100, respectively. The quantized attention model "VGG-aib-qt" further decreases the errors over "VGG-aib" by 0.18% and 0.69% on the two datasets. Compared with other VGG-backboned attention mechanisms, ours also achieves superior classification performance. Similarly, "WRN-aib" and "WRN-aib-qt" achieve the best results on CIFAR-10 and CIFAR-100. Visualization of attention maps for "VGG-aib", "VGG-aib-qt", "WRN-aib" and "WRN-aib-qt" are shown in Fig. 8.

4.2.2. Fine-grained recognition

Fine-grained recognition aims to differentiate similar subordinate categories of a basic-level category, *e.g.*, bird species (Wah et al., 2011). The annotations are usually provided by domain experts who can distinguish the subtle differences among the highly-confused subcategories. Compared with the generic image classification, fine-grained recognition would benefit more from finding informative regions that highlight the visual differences among the subordinate categories, and extracting discriminative features therein.

Datasets. CUB-200-2011 (CUB) (Wah et al., 2011) contains 5994 training and 5794 testing images from 200 bird species. The subtle differences among many of the species are sometimes even difficult for humans to distinguish. Street View Home Number (SVHN) (Netzer et al., 2011) collects 73,257 training, 26,032 testing, and 531,131 extra digit images from house numbers in street view images. The image resolution is 32×32. For CUB, we perform the preprocessing as Jetley et al. (2018). For SVHN, we apply the same data augmentation as CIFAR datasets.

Network configurations. For SVHN, the network configurations are the same as those of CIFARs, as presented in Table 1 and Section 4.1.1. For CUB with larger image resolution, the network structures are slightly different. As shown in Table 4, the feature extractor is built from the whole convolutional part of VGG16 or the first three blocks of WRN50-2. The encoder contains an FC layer for VGG backbone, and one block and an FC layer for WRN backbone. The structures of the attention module and the decoder are similar to those of CIFARs. For "VGG-aib-qt" and "WRN-aib-qt", the number of anchor values Q are 20 and 50, respectively. The anchor values $\{v_i\}$ are trained following the image classification task.

Training and inference. The training of "VGG-aib(-qt)" and "WRN-aib(-qt)" on SVHN is the same as those on CIFARs.

For CUB, "VGG-aib(-qt)" and "WRN-aib(-qt)" are all initialized with weights pretrained on ImageNet following Jetley et al. (2018). We use an Adam (Kingma and Ba, 2015) optimizer to fine-tune the model for 200 epochs. The initial learning rate is 10^{-4}. The training batch size is 16 and the test batch size is 4. We set $L_{\mathbf{a}} = L_{\mathbf{z}} = 1$ for training, and $L_{\mathbf{a}} = L_{\mathbf{z}} = 4$ for testing.

Results on CUB. As shown in Table 5, the proposed "VGG-aib" and "VGG-aib-qt" achieves smaller classification errors on CUB compared with baselines built on VGG, including GAP (Zhou et al., 2016), PAN (Seo et al., 2018), and VGG-att (Jetley et al., 2018). Specially, "VGG-aib" and "VGG-aib-qt" outperform VGG-att for a relative error decrease of 3.07% and 4.97%, respectively. Our "WRN-aib" and "WRN-aib-qt" also improve over the standard WRN for a noticeable margin. The visualizations for the CUB dataset (Column"CUB" in Fig. 8) demonstrate how different models attend to specific parts of bird images. Specifically, "VGG-aib" and "WRN-aib" show dispersed attention across various parts of the birds, highlighting features like wings, beaks, and

[1] https://github.com/SaoYan/LearnToPayAttention
[2] https://github.com/machine-perception-robotics-group/attention_branch_network

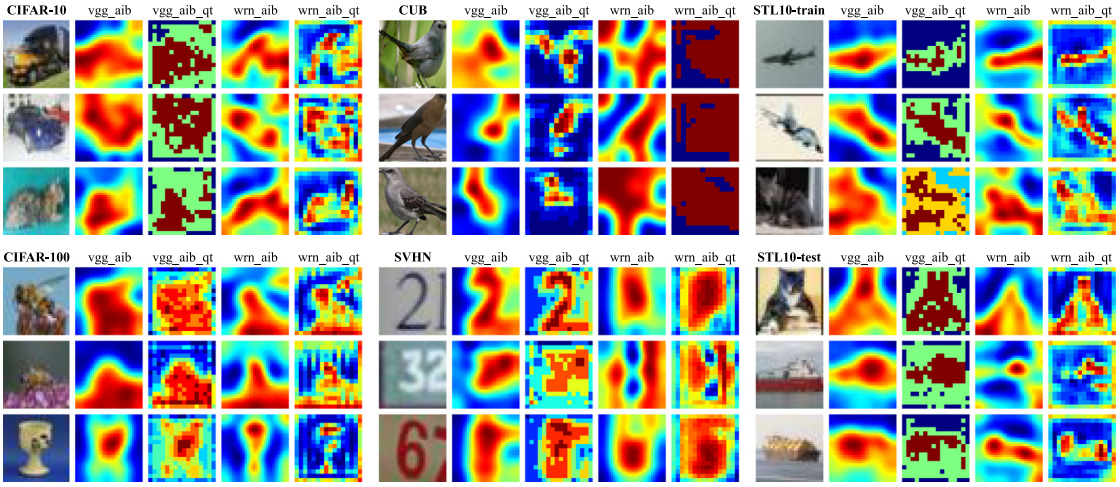


Fig. 8. Visualization of attention maps for various visual recognition tasks. This figure illustrates the attention maps for image classification (CIFAR-10, CIFAR-100), fine-grained classification (CUB, SVHN), and cross-dataset classification (STL10-train, STL10-test). See Sections 4.2.1–4.2.3 for details.

tails. On the other hand, "VGG-aib-qt" and "WRN-aib-qt" show a more focused and refined attention map, emphasizing the most discriminative parts of the birds. This tighter focus aligns with the goal of fine-grained classification, where subtle differences in specific parts are crucial for accurate categorization.

Results on SVHN. Our proposed method achieves lower errors with VGG-backboned models, and comparative errors for WRN-backboned methods. The visualizations for the SVHN dataset (Column"SVHN" in Fig. 8) demonstrate how different models process digit images from house numbers. Specifically, "VGG-aib" and "WRN-aib" produce attention maps that highlight various regions of the digit images, sometimes covering background areas as well as the digits themselves. With quantization, the attention maps of "VGG-aib-qt" and "WRN-aib-qt" become more concentrated around the digits, particularly focusing on the central parts of the numbers, thus improving digit recognition performance.

4.2.3. Cross-domain classification

Cross-domain classification is to test the generalization ability of a trained classifier on other domain-shifted benchmarks, *i.e.*, datasets that have not been used during training.

Datasets. STL-10 (Coates et al., 2011) is an image recognition dataset derived from ImageNet (Deng et al., 2009), which contains 100,000 unlabeled images, as well as 13,000 labeled images. These labeled images are categorized into 10 classes, with a division of 5000 for training and 8000 for testing. Each image within this dataset has a resolution of 96×96 pixels.

Training and inference. Following Jetley et al. (2018), models trained on CIFAR-10 are directly tested on the `train` and `test` sets of STL-10 without fine-tuning. Images in STL-10 are resized to 32×32 to fit the input resolution of the tested models.

Results on STL-10. As demonstrated in Table 5, the proposed attention model shows better generalization ability than other competitors with VGG backbone, and achieves comparative performance for those with WRN backbone. The visualizations for the STL-10 dataset are shown in Columns "STL10-train" and "STL10-test" of Fig. 8. Similar to the above visual recognition tasks, models with quantized attention tend to produce more concentrated attention maps.

4.2.4. Semantic segmentation

Semantic segmentation is a dense prediction task that assigns each pixel a particular label, yielding a pixel-wise prediction for each input image.

Datasets. ADE20K (Zhou et al., 2019) is a large scale semantic segmentation benchmark covering 150 semantic categories of various objects (*e.g.*, ball, bicycle, *etc.*) and stuff (*e.g.*, river, sky, *etc.*). There are 20k images designated for training and 2k images for validation purposes.

Network configurations. We extend the UperNet (Xiao et al., 2018) realized with the BEiT-base backbone (Bao et al., 2021) with our IB-inspired spatial attention mechanism, where we learn the attention scores for the visual tokens that naturally have spatial meaning. The input size is 512×512. The image patch size is 16×16. The number of visual tokens is 1024, and the resolution of the spatial attention map is 32×32. For the attention module, g_e^{μ} consists of a convolutional layer with 3×3 kernel followed by Sigmoid activation, and g_e^{σ} is simply a convolutional layer whose kernel size is 1×1. To realize the Gaussian encoder, the module right before the final segmentation head is adjusted to predict the mean and standard deviation of the latent feature, and the KL divergence loss is evaluated using Eq. (19). Other network structure settings such as the dimension of the embeddings are all consistent with those of the BEiT baseline. The decoder outputs $\mathbf{y} \in \mathbb{R}^{W \times H \times C}$ with spatial resolution $W \times H$ and channel number C, and C equals to the number of semantic classes.

Training and inference. Following Bao et al. (2021), we use Adam (Kingma and Ba, 2015) optimizer and set the initial learning rate as 10^{-3} with layer-wise decay. The model is initialized using the BEiT-base weight of Bao et al. (2021) that has been fine-tuned on ImageNet. We further train our model on ADE20K for 160k steps. The batch size is 16.[3] All other hyperparameters follow the settings of Bao et al. (2021). Our model is trained on 4 NVIDIA Titan Xp GPUs. We set $L_{\mathbf{a}} = L_{\mathbf{z}} = 1$ for training and $L_{\mathbf{a}} = 2, L_{\mathbf{z}} = 1$ during inference.

Results on ADE20K. We compare our model "BEiT-aib" with a recent semantic segmentation method DINO (Caron et al., 2021) as well as the original BEiT (Bao et al., 2021). Table 6 shows that our "BEiT-aib" outperforms DINO by a large margin, and is 2.06 points better than its counterpart without the spatial attention mechanism (*i.e.*, BEiT). "BEiT-aib-qt" further improves over "BEiT-aib" by 0.20 points. Fig. 9 visualizes the attention maps, showing how our models focus on different regions of the images to accurately segment objects. The quantized version "BEiT-aib-qt" shows more refined and focused attention than "BEiT-aib", which leads to improved performance.

4.2.5. Object detection

Object detection involves identifying and localizing objects in an image. It classifies objects into predefined categories and provides their spatial locations in the form of bounding boxes

[3] We use gradient accumulation to imitate the default batch size used in Bao et al. (2021) due to the limitation of the device.

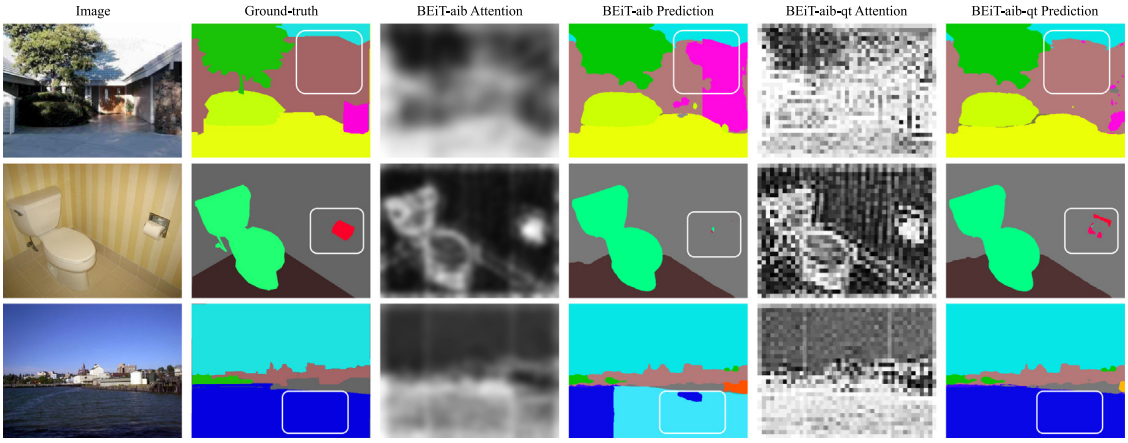


Fig. 9. Visualization of attention maps for semantic segmentation on ADE20K. The attention maps demonstrate how our models "BEiT-aib" and "BEiT-aib-qt" focus on different regions of the images to accurately segment various objects, with the quantized version "BEiT-aib-qt" showing more refined and focused attention, leading to improved segmentation performance. See Section 4.2.4 for details.

Table 6
Semantic segmentation results on ADE20K. Here, "aib" denotes "attentive information bottleneck" and "qt" denotes "quantization". Best value is in **bold**. See Section 4.2.4 for details.

Model	Input size	mIoU
DINO (Caron et al., 2021)	512 × 512	44.08
BEiT + Intermediate Fine-Tuning	512 × 512	47.70
BEiT-aib + Intermediate Fine-Tuning	512 × 512	49.76
BEiT-aib-qt + Intermediate Fine-Tuning	512 × 512	**49.96**

Table 7
Object detection results on MS COCO 2017. Here, "aib" denotes "attentive information bottleneck" and "qt" denotes "quantization". Best value is in **bold**. See Section 4.2.5 for details.

Model	mAP	mAP@50	mAP@75
Faster R-CNN (Ren et al., 2017)	37.4	58.3	41.2
Faster R-CNN-aib	38.3	59.4	41.4
Faster R-CNN-aib-qt	**38.4**	**59.3**	**41.7**

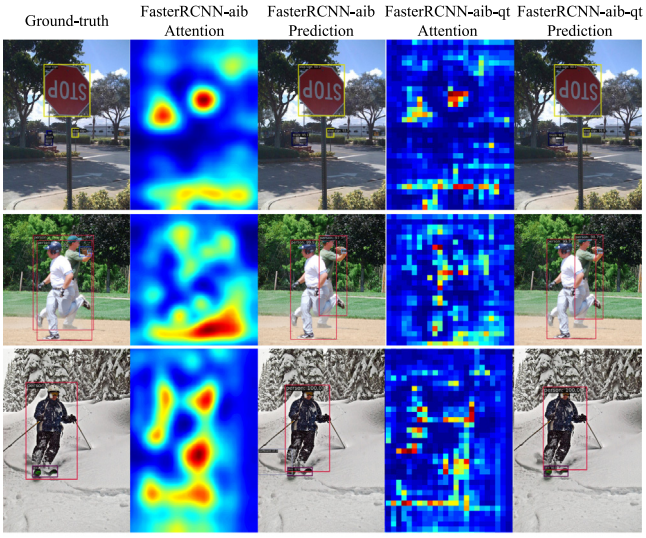


Fig. 10. Visualization of attention maps for the object detection task on MS COCO 2017. Our attention mechanism focuses on critical areas of the images to accurately detect objects. See Section 4.2.5 for details.

Datasets. MS COCO (Lin et al., 2014) is a large-scale object detection dataset containing bounding boxes and per-instance segmentation masks with 80 object categories. The 2017 training/validation split is 118k/5k.

Network configurations. We extend the Faster R-CNN (Ren et al., 2017) realized with the ResNet-50 backbone (He et al., 2016) with our IB-inspired spatial attention mechanism. Our implementation is based on the MMDetection framework (Chen et al., 2019). For the attention module, g_e^{μ} consists of a convolutional layer with 3×3 kernel followed by Sigmoid activation, and g_e^{σ} is simply a convolutional layer whose kernel size is 1×1. To realize the encoder, the module right before the final classification and regression heads is adjusted to predict the means and standard deviations of the latent features for the two heads, respectively. Other network structure settings such as the dimension of the embeddings are all consistent with those of the Faster R-CNN baseline.

Training and inference. We follow the default training and evaluation pipelines of mmDetection. Specifically, we use SGD optimizer and set the initial learning rate as 0.02 with momentum 0.9 and weight decay as 0.0001. We train our model for 12 epochs, and the learning rate is multiplied by 0.1 at the 8-th and 11-th epochs, respectively. The batch size is 16. We set $L_{\mathbf{a}} = L_{\mathbf{z}} = 1$ for training and $L_{\mathbf{a}} = 4, L_{\mathbf{z}} = 1$ during inference.

Results on MS COCO. We compare our model "Faster R-CNN-aib" with the original Faster R-CNN (Ren et al., 2017) under the same settings using mmDetection. Table 7 shows that our "Faster R-CNN-aib" is 0.9 points better than its counterpart without the spatial attention mechanism (*i.e.*, Faster R-CNN) regarding the Mean Average Precision (mAP) metric. "Faster R-CNN-aib-qt" further improves over "Faster R-CNN-aib" by 0.10 points. As shown in Fig. 10, our method focuses on critical areas of the images to accurately detect objects. The quantized attention maps exhibit more precise and concentrated attention, resulting in improved detection performance.

4.3. Interpretability assessment

In this section, we assess the interpretability of our attention maps relative to those of other attention models both qualitatively and quantitatively.

An interpretable attention map is one that accurately identifies and highlights the regions critical to the decision-making process of the model. Consequently, an interpretable attention mechanism is expected to produce consistent attention maps for an original input and its altered version, provided that the alternations do not affect the model decision. To quantitatively measure the interpretability of an attention mechanism, we compute an "interpretability score" as the proportion of attention-consistent samples in prediction-consistent samples under

Table 8
Interpretability scores on CIFAR-10 and CIFAR-100 datasets under spatial and frequency domain modifications. p represents the window size of the modified region. r indicates the radius in the frequency domain. See Section 4.3 for more details. * denotes re-implementation. Best values marked in **bold**.

Model	Spatial	CIFAR-10				CIFAR-100				Frequency	CIFAR-10	CIFAR-100
		$p=4$	$p=8$	$p=12$	$p=16$	$p=4$	$p=8$	$p=12$	$p=16$			
VGG-att3*	color	91.46	79.61	37.71	7.12	52.92	39.93	25.40	14.56	$r>4$	83.06	3.69
	svhn	90.97	75.70	36.74	6.51	82.08	63.16	39.07	21.03	$r<12$	49.61	50.90
VGG-aib	color	99.22	99.69	**93.78**	**20.59**	98.88	98.56	**95.92**	**22.70**	$r>4$	99.91	**99.78**
	svhn	98.59	97.52	97.35	72.86	98.73	96.70	96.69	93.02	$r<12$	53.26	79.13
VGG-aib-qt	color	**99.26**	**99.79**	91.10	18.36	**99.18**	**99.30**	94.59	20.54	$r>4$	**99.96**	99.60
	svhn	**98.65**	**98.04**	**97.85**	**78.82**	**99.12**	**97.64**	**97.21**	**94.79**	$r<12$	**73.52**	**79.70**
WRN-ABN	color	90.76	65.01	33.89	13.24	89.74	61.38	30.56	9.67	$r>4$	38.14	14.04
	svhn	90.40	68.41	38.46	16.55	92.77	63.67	30.57	9.47	$r<12$	36.33	25.43
WRN-aib	color	**99.95**	**93.94**	**45.17**	6.69	99.86	95.35	64.27	17.15	$r>4$	**78.96**	90.44
	svhn	**99.94**	**97.34**	**81.98**	**47.65**	99.90	97.77	**89.37**	62.64	$r<12$	**84.58**	**94.18**
WRN-aib-qt	color	99.84	84.18	28.94	4.64	**99.97**	**97.07**	**69.51**	**25.80**	$r>4$	72.05	**94.13**
	svhn	99.91	96.05	70.75	28.35	**99.95**	**98.52**	89.00	**63.10**	$r<12$	76.53	93.17

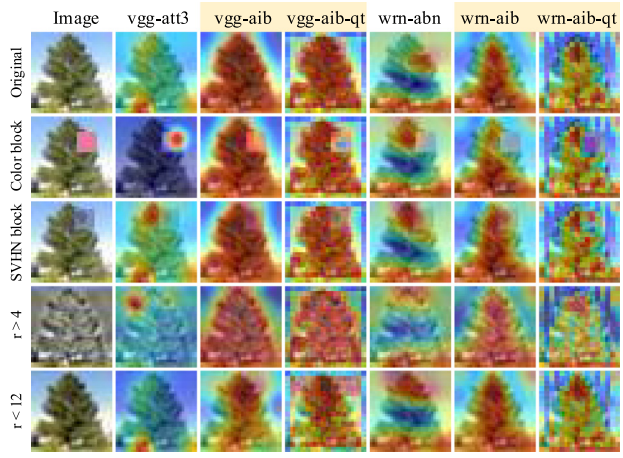


Fig. 11. Visualization of attention maps for interpretability analysis under different image modifications. See Section 4.3 for more details.

modifications either in the spatial or frequency domains. Attention consistency is determined by calculating the cosine similarity between two flattened and normalized attention maps.

Spatial domain modification analysis. We perform spatial domain modification by randomly occluding the original images from CIFAR datasets with either color blocks or images drawn from distinct datasets. The size of the occluded region p ranges from 4 to 16 pixels. The second and third rows of Fig. 11 show examples where imaged are occluded with a random color block and an image randomly drawn from SVHN dataset, respectively, where $p = 8$. Notably, our spatial attention model "VGG-aib(-qt)" and "WRN-aib(-qt)" produce attention maps that are consistent with those of the original images shown in the first row. The interpretability scores are detailed in Table 8. Our method consistently outperforms other spatial attention mechanisms using the same backbone across two datasets when the region size p increases from 4 to 16. This superior performance is attributed to the IB-inspired attention mechanism which effectively minimizes the MI between the attention-modulated features and the inputs, thereby reducing the impact of ambiguous information exerted on the inputs.

Frequency domain modification analysis. We also explore frequency domain modification, which involves applying a Fourier transform to the original images and selectively feeding only the high-frequency (HF) or low-frequency (LF) components into the model. To ensure sufficient information retention and maintain the classification performance, we focus on HF components with $r > 4$, and LF components with $r < 12$, where r is the radius in frequency domain as defined in Wang et al. (2020). The fourth and fifth rows of Fig. 11 illustrate exemplar images derived from HF and LF components, respectively. Our attention maps demonstrate greater fidelity to those of the original images compared with other competitors. Moreover, our method significantly outperforms other attention models in quantitative evaluations, as shown in Table 8.

4.4. Failure cases

In this section, we present failure cases of attention maps to analyze the limitations of our method.

Fig. 12 presents failure cases for different visual recognition tasks, including image classification (CIFAR-10, CIFAR-100), fine-grained classification (CUB, SVHN), and cross-dataset classification (STL10-train, STL10-test). Though our method achieves higher quantitative performance, there exist instances in which the models struggled to accurately focus on the most relevant parts of the images. This shows the limitations of our attention mechanisms in complex or ambiguous scenarios, providing insights into potential areas of improvement.

Fig. 13 illustrates failure cases in semantic segmentation. The failures primarily involve the inability to accurately capture small or ambiguous objects and incorrect segmentation boundaries for objects with complex shapes. These issues indicate that our models struggle with objects blending into their surroundings (such as the cushion on the chair) and fine details (such as the stair handrail) in complex scenes. Integrating a stronger feature extraction backbone could potentially enhance attention learning and help overcome these limitations.

Fig. 14 presents failure cases in object detection. A common type of failure observed is the tendency to predict more objects than are present in the ground-truth annotations. For instance, in the first row, the model incorrectly predicts the keyboard of the laptop separately. In the second row, the model detects more people than are annotated. This suggests a need for refining attention mechanisms in object classification to better distinguish between actual objects and spurious detections.

5. Discussion

5.1. Further explanation of attention quantization module

As shown in Tables 5, 6 and 7, the proposed attention quantization module can help the model achieve better performance. This section provides a further explanation of why attention quantization improves the performance of the model.

First, the quantization module can effectively reduce the redundant information processed by the model by rounding the attention scores to the nearest anchor values. This operation enhances the effect of the IB-inspired attention mechanism by focusing on the most relevant information that aligns with the principles of the Human Visual System,

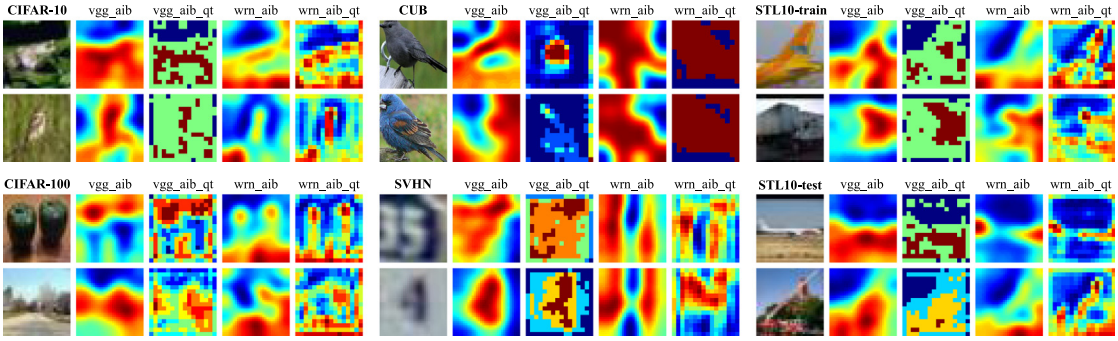


Fig. 12. Failure cases for various visual recognition tasks. Our method struggles to accurately focus on the most relevant parts of the images with complex or ambiguous scenarios. See Section 4.4 for details.

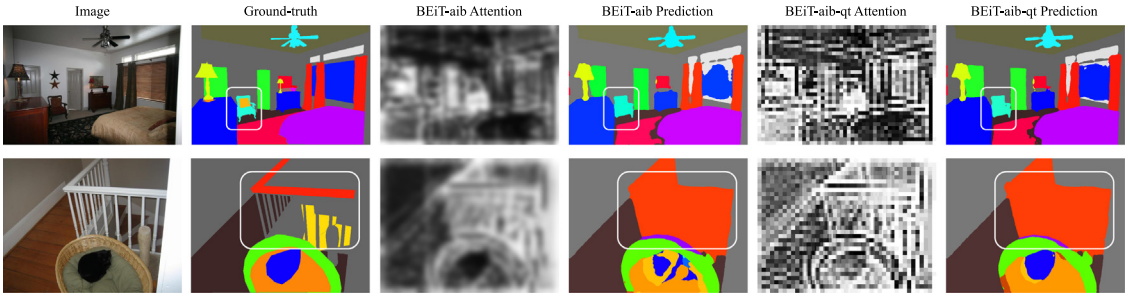


Fig. 13. Failure cases for the semantic segmentation task on ADE20K. Our models struggle with objects blending into their surroundings (such as the cushion on the chair) and fine details (such as the stair handrail) in complex scenes. See Section 4.4 for details.

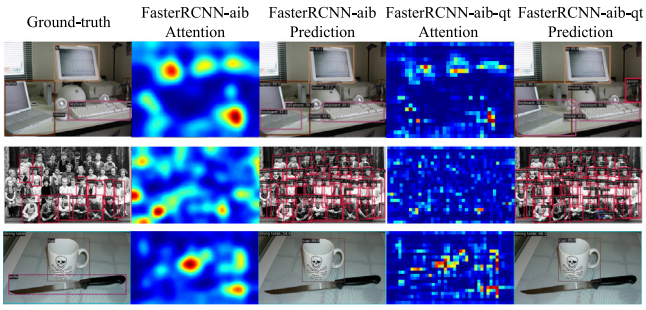


Fig. 14. Failure cases for object detection task on MS COCO 2017. In some cases, our model tends to predict more objects than are present in the ground-truth annotations. See Section 4.4 for details.

thereby improving the efficiency and accuracy of the model. Second, by constraining the attention scores to a limited set of anchor values, the model learns to generalize better across different tasks and datasets. This is particularly beneficial in preventing overfitting and ensuring robust performance in various scenarios. Third, the quantization module provides more interpretable attention maps by applying further constraints on the attention scores, as evidenced in Sections 4.3 and 4.1.2. It can further help the model focus on more concentrated regions with important information, thus making it easier to understand what regions of the input the model focuses on. The visualization results in Figs. 8, 9, and 10 also provide additional qualitative evidence to support the quantitative findings.

5.2. Integration with advanced backbones

Integrating our IB-inspired attention mechanism with newer architectures offers significant potential benefits. Recent advancements in backbone architectures, such as Swin Transformer (Liu et al., 2021), DNC (Wang et al., 2023), and Mamba (Gu and Dao, 2023), provide enhanced feature extraction capabilities. By leveraging these developments, our approach can potentially achieve better performance, scalability, and generalization across various tasks. However, challenges include the need for substantial computational resources to train and optimize these advanced models and ensure compatibility and seamless integration of our attention mechanism with the complex structures of newer architectures. Addressing these challenges requires careful adaptation to harness the full potential of combining our method with cutting-edge backbones.

6. Conclusion

We propose a spatial attention mechanism inspired by IB theory, which is designed to generate probabilistic maps that minimize the MI between the masked representation and the input while maximizing MI between the masked representation and the task label. To enhance flexibility, an adversarial learning strategy is incorporated into the framework. Additionally, to further restrict the information bypassed by the attention map, we introduce an adaptive quantization mechanism that regularizes the attention scores by rounding continuous score values to the nearest anchor value during training. Extensive experiments demonstrate that this IB-inspired spatial attention mechanism under various configurations significantly improves the performance in visual recognition and dense prediction tasks by concentrating on the most informative parts of the input. The resulting attention maps provide interpretable insights into the decision-making of the DNNs, as they consistently highlight the informative regions across the original and modified inputs with identical semantics.

CRediT authorship contribution statement

Qiuxia Lai: Writing – original draft, Visualization, Validation, Methodology, Investigation, Funding acquisition, Formal analysis. **Yongwei Nie:** Writing – review & editing, Supervision, Resources, Project administration, Methodology, Funding acquisition, Formal

analysis. **Yu Li:** Writing – review & editing, Visualization, Validation, Software, Methodology, Investigation. **Hanqiu Sun:** Writing – review & editing, Validation, Supervision, Methodology, Formal analysis. **Qiang Xu:** Writing – review & editing, Supervision, Resources, Methodology, Funding acquisition, Formal analysis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

In this paper, we utilize public datasets. The code is available at https://github.com/ashleylqx/AIB-Ex.git.

Acknowledgments

This work is sponsored by the National Natural Science Foundation of China (No. 62072191, 62306292) and the Fundamental Research Funds for the Central Universities, China (No. CUC24QT06, D2240210).

References

Achille, A., Soatto, S., 2018. Information dropout: Learning optimal representations through noisy computation. TPAMI 40 (12), 2897–2905.
Alcazar, J.L., Bravo, M.A., Jeanneret, G., Thabet, A.K., Brox, T., Arbelaez, P., Ghanem, B., 2021. MAIN: Multi-attention instance network for video segmentation. CVIU 210, 103240.
Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K., 2017. Deep variational information bottleneck. In: ICLR.
An, D., Wang, H., Wang, W., Wang, Z., Huang, Y., He, K., Wang, L., 2024. Etpnav: Evolving topological planning for vision-language navigation in continuous environments. IEEE TPAMI.
Bahdanau, D., Cho, K., Bengio, Y., 2015. Neural machine translation by jointly learning to align and translate. In: ICLR.
Bao, H., Dong, L., Wei, F., 2021. BEiT: BERT pre-training of image transformers. ArXiv preprint arXiv:2106.08254.
Belghazi, M.I., Baratin, A., Rajeshwar, S., Ozair, S., Bengio, Y., Courville, A., Hjelm, D., 2018. Mutual information neural estimation. In: ICML. pp. 531–540.
Bengio, Y., Léonard, N., Courville, A., 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. ArXiv preprint arXiv:1308.3432.
Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging properties in self-supervised vision transformers. ArXiv preprint arXiv:2104.14294.
Chen, J., Gao, C., Meng, E., Zhang, Q., Liu, S., 2022. Reinforced structured state-evolution for vision-language navigation. In: CVPR. pp. 15450–15459.
Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C.C., Lin, D., 2019. MMDetection: Open MMLab detection toolbox and benchmark. ArXiv preprint arXiv:1906.07155.
Chen, Y., Xia, R., Yang, K., Zou, K., 2024. MFMAM: Image inpainting via multi-scale feature module with attention module. CVIU 238, 103883.
Chorowski, J.K., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y., 2015. Attention-based models for speech recognition. In: NeurIPS.
Coates, A., Ng, A., Lee, H., 2011. An analysis of single-layer networks in unsupervised feature learning. In: AISTATS.
Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database. In: CVPR.
Eriksen, C.W., Hoffman, J.E., 1972. Temporal and spatial characteristics of selective encoding from visual displays. Percept. Psychophys. 12 (2), 201–204.
Fukui, H., Hirakawa, T., Yamashita, T., Fujiyoshi, H., 2019. Attention branch network: Learning of attention mechanism for visual explanation. In: CVPR.
Gao, C., Liu, S., Chen, J., Wang, L., Wu, Q., Li, B., Tian, Q., 2023. Room-object entity prompting and reasoning for embodied referring expression. IEEE TPAMI.
Gu, A., Dao, T., 2023. Mamba: Linear-time sequence modeling with selective state spaces. ArXiv preprint arXiv:2312.00752.
Guo, M.-H., Xu, T.-X., Liu, J.-J., Liu, Z.-N., Jiang, P.-T., Mu, T.-J., Zhang, S.-H., Martin, R.R., Cheng, M.-M., Hu, S.-M., 2021. Attention mechanisms in computer vision: A survey. ArXiv preprint arXiv:2111.07624.
Han, K., Wang, Y., Chen, H., et al., 2020. A survey on visual transformer. ArXiv preprint arXiv:2012.12556.
He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: CVPR. pp. 770–778.
Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y., 2019. Learning deep representations by mutual information estimation and maximization. In: ICLR.
Hu, J., Shen, L., Sun, G., 2018. Squeeze-and-excitation networks. In: CVPR.
Hu, H., Zhang, Z., Xie, Z., Lin, S., 2019. Local relation networks for image recognition. In: ICCV.
Hui, T., Liu, S., Ding, Z., Huang, S., Li, G., Wang, W., Liu, L., Han, J., 2023. Language-aware spatial-temporal collaboration for referring video segmentation. IEEE TPAMI.
Jetley, S., Lord, N.A., Lee, N., Torr, P.H., 2018. Learn to pay attention. In: ICLR.
Kingma, D., Ba, J., 2015. Adam: A method for stochastic optimization. In: ICLR.
Kingma, D.P., Welling, M., 2014. Auto-encoding variational bayes. In: ICLR.
Koch, K., McLean, J., et al., 2006. How much the eye tells the brain. Curr. Biol. 16 (14), 1428–1434.
Krizhevsky, A., Hinton, G., et al., 2009. Learning Multiple Layers of Features from Tiny Images. Technical Report.
Lai, Q., Li, Y., Zeng, A., Liu, M., Sun, H., Xu, Q., 2021. Information bottleneck approach to spatial attention learning. In: IJCAI.
Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In: ECCV. pp. 740–755.
Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B., 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022.
Makhzani, A., Shlens, J., Jaitly, N., Goodfellow, I., Frey, B., 2016. Adversarial autoencoders. In: ICLR Workshop.
Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y., 2011. Reading digits in natural images with unsupervised feature learning. In: NeurIPS Workshop.
Qin, J., Bai, H., Zhao, Y., 2021. Multi-scale attention network for image inpainting. CVIU 204, 103155.
Ren, S., He, K., Girshick, R., Sun, J., 2017. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE TPAMI.
Ren, P., Li, C., Wang, G., Xiao, Y., Du, Q., Liang, X., Chang, X., 2022. Beyond fixation: Dynamic window visual transformer. In: CVPR. pp. 11987–11997.
Schulz, K., Sixt, L., Tombari, F., Landgraf, T., 2019. Restricting the flow: Information bottlenecks for attribution. In: ICLR.
Seo, P.H., Lin, Z., Cohen, S., Shen, X., Han, B., 2018. Progressive attention networks for visual attribute prediction. In: BMVC.
Sharma, S., Kiros, R., Salakhutdinov, R., 2015. Action recognition using visual attention. In: ICLR Workshop.
Simonyan, K., Vedaldi, A., Zisserman, A., 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. ArXiv preprint arXiv:1312.6034.
Simonyan, K., Zisserman, A., 2015. Very deep convolutional networks for large-scale image recognition. In: ICLR.
Sun, Q., Li, J., Peng, H., Wu, J., Fu, X., Ji, C., Philip, S.Y., 2022. Graph structure learning with variational information bottleneck. In: AAAI, vol. 36, no. 4. pp. 4165–4174.
Taghanaki, S.A., Havaei, M., Berthier, T., Dutil, F., Di Jorio, L., Hamarneh, G., Bengio, Y., 2019. Infomask: Masked variational latent representation to localize chest disease. In: MICCAI.
Tishby, N., Pereira, F.C., Bialek, W., 1999. The information bottleneck method. JMLR.
Van Den Oord, A., Vinyals, O., et al., 2017. Neural discrete representation learning. In: NeurIPS.
Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: NeurIPS.
Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S., 2011. The Caltech-UCSD Birds-200–2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology.
Wang, W., Han, C., Zhou, T., Liu, D., 2023. Visual recognition with deep nearest centroids. In: ICLR.
Wang, W., Lu, X., Shen, J., Crandall, D.J., Shao, L., 2019a. Zero-shot video object segmentation via attentive graph neural networks. In: ICCV. pp. 9236–9245.
Wang, W., Song, H., Zhao, S., Shen, J., Zhao, S., Hoi, S.C., Ling, H., 2019b. Learning unsupervised video object segmentation through visual attention. In: CVPR. pp. 3064–3074.
Wang, H., Wu, X., Huang, Z., Xing, E.P., 2020. High-frequency component helps explain the generalization of convolutional neural networks. In: CVPR.
Wang, W., Zhao, S., Shen, J., Hoi, S.C., Borji, A., 2019c. Salient object detection with pyramid attention and salient edges. In: CVPR. pp. 1448–1457.
Woo, S., Park, J., Lee, J.-Y., So Kweon, I., 2018. Cbam: Convolutional block attention module. In: ECCV.
Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J., 2018. Unified perceptual parsing for scene understanding. In: ECCV. pp. 418–434.
Xu, K., Ba, J., et al., 2015. Show, attend and tell: Neural image caption generation with visual attention. In: ICML.
Yuan, H., Sun, Q., Fu, X., Ji, C., Li, J., 2024. Dynamic graph information bottleneck. In: WWW. pp. 469–480.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks. In: BMVC.
Zhmoginov, A., Fischer, I., Sandler, M., 2019. Information-bottleneck approach to salient region discovery. In: ICML Workshop.
Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A., 2016. Learning deep features for discriminative localization. In: CVPR.
Zhou, T., Porikli, F., Crandall, D.J., Van Gool, L., Wang, W., 2022. A survey on deep learning technique for video segmentation. IEEE TPAMI 45 (6), 7099–7122.
Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., 2019. Semantic understanding of scenes through the ade20k dataset. IJCV 127 (3), 302–321.