Self Supervised Learning Visual Encoder with Mask Aware Gating

Dr. Hemachandran K*

*Professor, AI Research Centre, School of Business, Woxsen University, India; Email: hemachandran.k@woxsen.edu.in

*Corresponding Author: hemachandran.k@woxsen.edu.in

DOI: https://doi.org/10.30211/JIC.202503.017

Submitted: Oct. 24, 2025 Accepted: Dec. 05, 2025

ABSTRACT

In the context of self-supervised learning, traditional visual encoders are often limited in processing complex scenes due to the lack of effective feature selection mechanisms. This article proposes a self-supervised learning visual encoder model that combines a mask-aware gating mechanism to enhance the feature learning ability in visual tasks. Specifically, we simulated data loss or noise interference by partially masking input images and designed a hyper self-aware gating module (HSAGM) that adaptively adjusts the model's attention to different features, improving the efficiency and accuracy of feature extraction. In addition, we proposed a mask-based interpolation method that utilizes contextual information learned by the model to make reasonable interpolation predictions for occluded areas. These methods perform well in visual neural coding tasks, improving the training performance of self-supervised learning models. Through experimental verification on benchmark datasets, our method significantly improves the performance of visual tasks such as image classification, demonstrating its potential for application in complex visual scenes.

Keywords: Visual encoder, Feature extraction, Self-supervised learning, Mask-aware.

1. Introduction

The visual neural coding model is a core tool for exploring and understanding how the brain processes visual information. By simulating neuronal responses, these models help reveal the internal mechanisms by which the brain receives and processes visual stimuli. They play an important role in computational neuroscience and artificial intelligence, enabling applications in image recognition [1], computer vision [2], and brain–computer interfaces [3]. However, as the scale and complexity of visual data continue to grow, traditional encoding models struggle with both efficiency and accuracy when processing high-dimensional information. Thus, developing more scalable, efficient, and biologically inspired visual neural encoding models has become a key research challenge.

Self-supervised learning (SSL) [4] has recently emerged as a powerful paradigm for automatically extracting visual features without requiring large annotated datasets. In visual neural encoding, convolutional neural networks (CNNs) [5] are widely used for spatial feature extraction, demonstrating strong performance in classification and detection tasks, though they remain limited

when dealing with high-dimensional or complex visual scenes. Recurrent neural networks (RNNs) [6], while capable of capturing temporal dynamics, suffer from vanishing gradients when processing long sequences. Autoencoders [7] support dimensionality reduction and robust representation learning, but their reconstruction bias limits their ability to encode rich structural information. Although SSL-based encoding frameworks [7][8] have achieved promising results, current methods still lack efficient temporal modeling capabilities and struggle to handle incomplete, noisy, or partially missing visual inputs conditions that frequently arise in real-world scenarios.

To address these limitations, this study aims to develop a self-supervised visual neural encoding framework capable of extracting stable spatiotemporal representations even under incomplete or degraded visual inputs**.** The core idea is to leverage contrastive learning to build a feature space in which similar visual signals lie closer together while dissimilar ones are separated. CNNs are used to extract spatial features, while contrastive objectives refine the embedding quality. However, biological visual processing is inherently spatiotemporal, and thus a complete neural encoding model must incorporate both spatial feature extraction and temporal dynamics.

This motivates the integration of pose-based temporal modeling as a downstream application scenario. Sports movements provide a natural testbed for evaluating the model's ability to encode complex visual dynamics, as they involve rapid body-joint interactions, occlusions, and noisy visual conditions closely mirroring the challenges encountered in visual neural encoding research. Therefore, pose estimation is not a departure from neural encoding; it serves as a practical, high-complexity benchmark task for validating the robustness of the proposed encoding architecture.

In this context, we incorporate EfficientPose for frame-level spatial feature extraction and T-GCN for temporal graph modeling, enabling the model to evaluate whether the learned neural representations support accurate, stable spatiotemporal understanding**.** EfficientPose generates high-precision joint representations, and T-GCN models temporal dependencies across frames, making the framework suitable for analyzing dynamic actions such as those in sports events.

Based on these motivations, the contributions of this work are summarized as follows:

- We designed a visual masking mechanism to simulate real-world data-loss scenarios, enabling the encoder to learn robust neural representations from incomplete visual inputs in a self-supervised manner.
- We proposed a hyper self-aware gating module that adaptively adjusts attention to input features, improving selective feature emphasis and enabling context-aware interpolation for missing or ambiguous visual regions.
- We further evaluated the proposed encoding model through a challenging downstream task – sports action pose estimation using EfficientPose for spatial encoding and T-GCN for temporal continuity. This demonstrates how the learned neural representations generalize beyond static visual encoding into dynamic spatiotemporal tasks.
- The proposed system is validated under both supervised and self-supervised settings, and each component is independently verified to illustrate its contribution.

2. Related Work

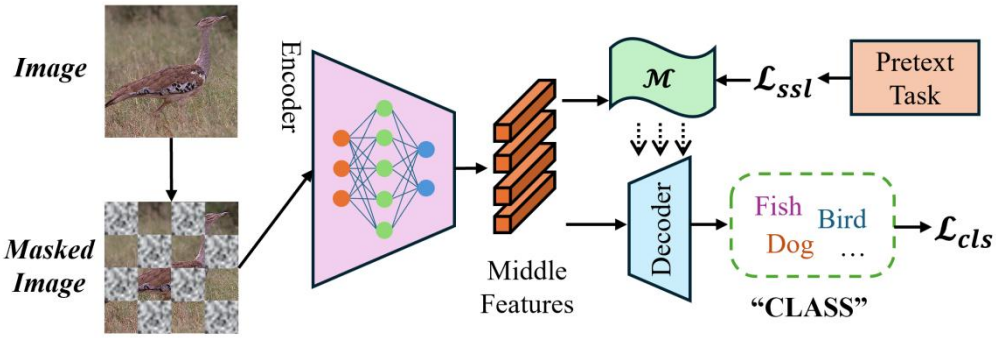

Figure 1. Self supervised learning (SSL) architecture

Figure 1. Self supervised learning (SSL) architecture, where the original image is masked and input into an encoder to extract intermediate features. Subsequently, the self supervised pre training task L_{ssl} is executed through the feature processing module, and the classification task L_{cls} is performed through the decoder, thereby improving the model's feature learning ability and classification performance.

2.1 Self-Supervised Learning

The field of computer vision has witnessed considerable advancement in the domain of self-supervised learning (SSL), a paradigm that eliminates reliance on manually labeled data [9]. SSL enables models to learn rich and transferable feature representations by designing pretext tasks that leverage intrinsic data structures. Among these, contrastive learning [10] has become one of the most influential approaches, as it learns discriminative features by maximizing similarity between positive pairs and minimizing similarity between negative pairs. Methods such as SimCLR [11], MoCo [12], and BYOL [13] achieve powerful representation learning through large-batch contrastive objectives or momentum encoders.

Moreover, autoencoders and generative adversarial networks (GANs) [14] play important roles in SSL. Autoencoders compress input data into a compact latent space and reconstruct it, thereby learning robust features for denoising and dimensionality reduction tasks. GANs generate realistic samples via adversarial training, implicitly learning the underlying data distribution. These techniques have proven effective in image generation, reconstruction, and clustering.

However, existing SSL frameworks generally assume complete visual inputs and struggle when data contain occlusions, noise, or missing regions – conditions prevalent in natural scenes and

biological visual processing. This limitation motivates the need for a more flexible visual encoding module capable of handling incomplete inputs while preserving semantic integrity.

2.2 Visual Neural Encoding

Visual neural coding aims to construct deep learning architectures inspired by biological visual systems, enabling the extraction and interpretation of visual information [15]. Convolutional neural networks (CNNs) remain foundational for this task due to their ability to learn hierarchical visual features from edges and textures to semantic concepts. Their success in classification, detection, and segmentation demonstrates their strong spatial feature-learning capacity.

Transformer-based visual models such as Vision Transformer (ViT) [16] extend this capability by viewing images as sequences of patches and applying global self-attention operations. While ViT overcomes the locality constraints of CNN receptive fields, it suffers from increased computational cost, especially for high-resolution images. Swin Transformer [17] improves efficiency via shifted window attention, enabling scalable feature modeling but sometimes losing fine-grained local information at window boundaries.

To address these challenges, enhanced attention mechanisms such as Hybrid Sparse Attention and Convolution-Augmented Multi-Head Attention (CAMHA) [18] have been proposed, though they introduce high tuning complexity and often require large, clean datasets. Their limited robustness under noisy or partially missing data restricts their usability in real-world visual tasks.

In contrast, the proposed Mask-Aware Gating (MAG) mechanism offers greater adaptability and noise tolerance. By explicitly incorporating a visual mask structure:

- the model learns to adjust its attention distribution across image regions,
- masked or unreliable regions receive down-weighted attention, and
- important contextual regions are emphasized.

MAG therefore enhances the model's resilience to incomplete visual inputs -- a property essential for self-supervised neural encoding and downstream temporal tasks such as pose estimation.

2.3 Pose Estimation Based on Graph Neural Networks

Graph neural networks (GNNs) have gained substantial attention in pose estimation because human skeletal structures naturally form graph topologies. Each joint corresponds to a node, and bones represent edges. This structural representation enables GNNs to capture global spatial dependencies that traditional CNN-based methods fail to model effectively.

In CNN-based pose estimation, joint positions are usually inferred from localized feature maps. While effective for simple poses, CNNs struggle with:

- long-range dependencies between distant joints,
- occlusions,
- multi-person interactions, and
- non-rigid body dynamics.

GNN-based methods, such as Graph-based Pose Estimation (GPE) by Size Wu et al., 2022 explicitly encode joint adjacency relationships, allowing the model to reason over the full skeletal

structure. GCN layers propagate information across connected joints, improving robustness in cluttered or dynamic environments.

Similarly, **Pose-GCN** by Ma et al. 2025 demonstrates that multi-level graph convolution improves both local and global reasoning. By stacking hierarchical graph layers, Pose-GCN captures:

- fine-grained local interactions (e.g., wrist–elbow),
- mid-level relationships (e.g., arm–torso), and
- holistic body dynamics.

These characteristics make GNNs ideal for downstream evaluation of the proposed visual neural encoding model. Since pose estimation is inherently spatiotemporal and sensitive to partial occlusions, it serves as an effective benchmark for testing the robustness of MAG-enhanced self-supervised visual representations.

3. Method

3.1. Self-Supervised Learning Theory

Self-supervised learning (SSL) utilizes the inherent structure of data itself for learning without manual annotation [19]. As shown in Figure 1, the basic principle is to design auxiliary tasks (pretext tasks) to mine its own supervised information from large-scale unsupervised data, in order to train the model. This learning method does not require external labeled data, but generates labels from the data itself. The goal of self-supervised learning is to learn valuable representations for downstream tasks, that is, to train a model using unlabeled data and then apply it to specific supervised learning tasks through transfer learning. In self-supervised learning, the model learns the intrinsic structure of the input data by predicting a certain transformation of the data.

A common method of SSL is to train models through contrastive learning. In contrastive learning, the goal of the model is to bring similar data samples closer while pushing dissimilar samples further away. Specifically, given a sample x_i and its positive x_j^+ and negative x_k^- samples, the loss function of contrastive learning is usually defined as in (1).

$$\mathcal{L}_{i,j} = -\log \frac{\exp(\sin(z_i, z_j^+))}{\exp(\sin(z_i, z_j^+)) + \sum_{k=1}^{K} \exp(\sin(z_i, z_k^-))} \cdots\cdots \text{[Formular 1]}$$

where z_i represents the representation of sample x_i, and $\mathrm{sim}(z_i, z_j)$ represents the similarity function between them, usually using residual similarity or dot product. The core idea is to maximize similarity between similar samples (i.e. x_i and x_j^+) while minimizing similarity between dissimilar samples. By minimizing the loss function, the model can learn feature representations that can effectively distinguish different samples.

3.2. Visual Mask Structure

We propose a Visual Masking Structure (VMS) that aims to enhance the model's focus on specific features by selectively masking certain parts of the image.The core principle of VMS is to dynamically adjust the weights of various regions in the input image by applying a mask operation, which enables the model to more effectively extract and utilize important visual information. This method is especially suitable for dealing with complex scenes or images with background interference,

such as in target detection, image segmentation and pose estimation, etc. VMS can significantly improve the accuracy and robustness of the model.

The mathematical representation of the visual mask structure can be described by the following equation as in (2).

$$\mathbf{I}_{masked} = \mathbf{I} \odot \mathbf{M} + (1 - \mathbf{M}) \odot \mathbf{N} \cdots\cdots\cdots\cdots\cdots \text{[Formular 2]}$$

where $\mathbf{I}_{masked}$ represents the masked image, $\mathbf{I}$ is the original input image, $\mathbf{M}$ is the mask matrix, $\mathbf{N}$ is the noisy image or padding value, $\odot$ representing element wise multiplication operation. Through this equation, the mask matrix $\mathbf{M}$ determines which pixels in the image will be retained and used by the model, while $(1 - \mathbf{M})$ determines the fill value of the masked area. This operation can effectively reduce the influence of background information and focus the model on the most informative part of the image.

Furthermore, in order to optimize the generation of masks, attention mechanisms or learned weights can be considered. If the attention distribution map generated by the model is $\mathbf{A}$ then the mask matrix $\mathbf{M}$ can be generated through the following steps as in (3).

$$\mathbf{M} = \tanh(\mathbf{W}_2 \cdot ReLU(\mathbf{W}_1 \cdot \mathbf{A} + b_1) + b_2) \cdots\cdots\cdots \text{[Formular 3]}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ are learnable weight matrices, b_1 and b_2 are bias vectors, $ReLU$ is the activation function, $\tanh$ represents the hyperbolic tangent function, used to map the output values to a [-1,1] interval. The multi-layer perceptron (MLP) architecture in this formula enables the model to adaptively generate a mask matrix $\mathbf{M}$ thereby optimizing the weight allocation of each region in the image. Through this method, the code matrix can not only adaptively adjust based on the global feature distribution of the image, but also further improve the performance of the model in specific tasks.

In addition to the basic principles of mask generation, visual mask structures can also be combined with deep CNN and RNN to further enhance their effectiveness. In this case, the mask matrix \$\mathbf{M}\$ can be generated by a complex neural network in the following form as in (4).

$$\mathbf{M} = \text{Sigmoid}\left(\sum_{l=1}^{L} \mathbf{W}_l * ReLU(\mathbf{F}_{l-1}) + b_l\right) \cdots\cdots\cdots \text{[Formular 4]}$$

Among them, $\mathbf{F}$ represents the feature map of layer $l-1$ of the network, $\mathbf{M}$ represents the convolution kernel of layer l, b is the bias vector, and $*$ represents the convolution operation. Through this formula, the mask matrix $\mathbf{M}$ can be generated by multi-layer convolution and nonlinear transformation, thereby capturing high-level features and local information of the image. The application of the Sigmoid function ensures that the values of the mask matrix are compressed within the range of [0,1] to accommodate element wise multiplication operations.

The visual mask structure, through a complex mask generation process, enables the model to dynamically and intelligently mask certain regions in the input image, thereby improving the ability to extract key features.

3.3. Hyper Self Aware Gate Control Module

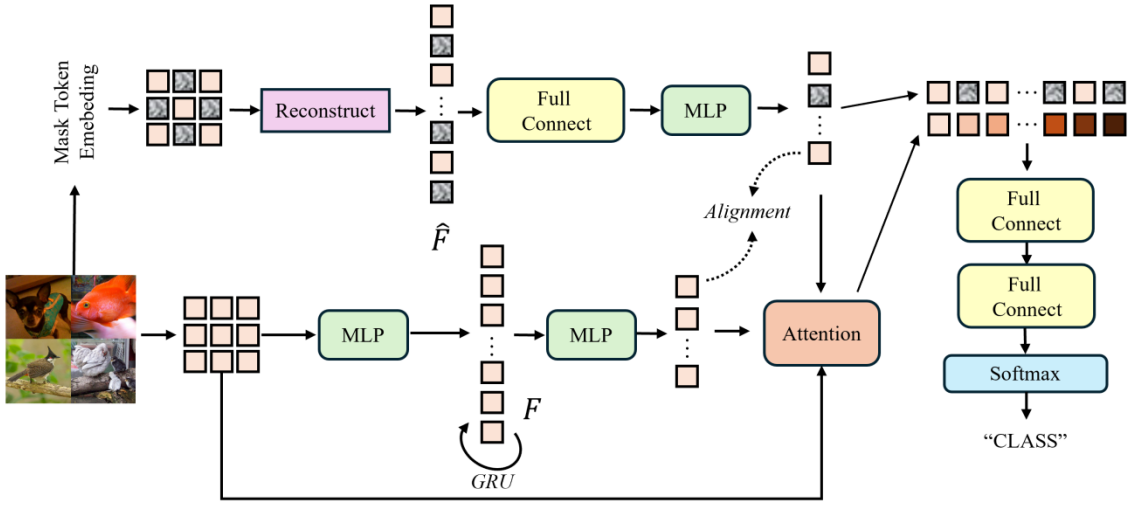

Figure 2. Architecture diagram of HSAGM

Figure 2. Architecture diagram of HSAGM. The input image is first masked to generate masked image blocks. After the image block is processed by a multi-layer perceptron (MLP), the feature representation F is extracted and dynamically aggregated by a gated recurrent unit (GRU). The aggregated feature F is aligned with the feature $\hat{F}$ generated by the reconstruction module, and the expression of key features is further enhanced through attention mechanism. Finally, the fully connected layer and Softmax classifier output the final classification result. This architecture utilizes a combination of masking and gating mechanisms, effectively improving the model's feature extraction ability and classification performance in complex scenes.

The hyper self aware gating module (HSAGM) aims to enhance the ability to learn masks through dynamic perception and gating mechanisms. In computer vision tasks, HSAGM optimizes image information through complex feature alignment and attention mechanisms. The network architecture is shown in Figure 2.

The working principle of HSAGM can be divided into two main steps: feature extraction and gating mechanism. Firstly, the input image is processed through a series of convolutional layers to extract preliminary features, which are then fed into an MLP network for further processing. Specifically, the input feature matrix F undergoes a nonlinear transformation through an MLP layer, and the output feature representation is F', as in (5).

$$F' = ReLU(W_1 \cdot F + b_1) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{[Formular 5]}$$

where, W_1 and b_1 respectively represent the weight matrix and bias vector of the first layer MLP. The ReLU function is used as an activation function to ensure non-linear transformation of features. Next, these features will be input into the GRU unit for gating processing. Through the updating and resetting gates of the GRU, the model can dynamically adjust the degree of fusion between the current state and the previous state, and finally output the updated feature F'. The calculation process of GRU is as in (6)(7)(8).

$$z = \sigma(\mathbf{W}_z \cdot [h_{t-1}, F'] + b_z) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \text{[Formular 6]}$$

$$r = \sigma(\mathbf{W}_r \cdot [h_{t-1}, F'] + b_r) \cdots\cdots\cdots\cdots\cdots\cdots \text{[Formular 7]}$$

$$h_t = z \odot h_{t-1} + (1 - z) \odot \tanh(\mathbf{W}_h \cdot [r \odot h_{t-1}, F'] + b_h)$$
$$\cdots\cdots\cdots\cdots\cdots\cdots \text{[Formular 8]}$$

Among them, z is the update gate, r is the reset gate, h_t is the feature representation in the current state, σ is the Sigmoid activation function. GRU effectively controls the flow of information between the front and back states through gating mechanisms, avoiding the gradient vanishing problem in traditional RNNs and enhancing the time series correlation of features. After feature extraction and gating processing, HSAGM introduces an alignment mechanism. The core goal of this mechanism is to achieve alignment between multiple layers of features, ensuring that the model can focus on the most relevant feature regions. Specifically, the alignment mechanism adjusts the weights of features through attention distribution maps. Set the feature mapping to $\hat{F}$, the calculation is as in (9).

$$\hat{F} = Softmax(\mathbf{W}_a \cdot ReLU(\mathbf{W}_b \cdot F'' + b_b) + b_a) \cdot \text{[Formular 9]}$$

where $\mathbf{W}_a$, $\mathbf{W}_b$ are the weight matrices in the attention mechanism, and b_a, b_b are the corresponding bias vectors. The Softmax function is used to normalize the weights to ensure that the sum of the weights of all the features is 1. In this way, the model can adaptively adjust the weight of the feature distribution to ensure that the most critical image information is retained during the feature alignment process.

3.4. Mask Based Interpolation Method

We propose a mask based interpolation method to compensate for the coordinate loss caused by masks during feature extraction. Introducing context aware feature reconstruction capability using KNNImputer's nearest neighbor feature value imputation mechanism to ensure imputation accuracy and effectiveness in complex scenes. In this combined methodology, a mask decoder is first used to label and reconstruct the feature representations of missing data. The core task of the mask decoder is to represent the missing value positions Mij in the data matrix $\mathbf{X}$ as a binary mask matrix Mij=1 indicates a missing value, while Mij=0 indicates no missing value. The mask decoder generates a reconstructed feature matrix $\mathbf{X}$ by learning the global context in the data, which is represented as in (10).

$$\widehat{\mathbf{X}} = Decoder(\mathbf{X} \odot (1 - \mathbf{M}), \mathbf{M}) \cdots\cdots\cdots\cdots\cdots \text{[Formular 10]}$$

where, $Decoder$ is a function of a mask decoder. This step aims to generate a preliminary candidate feature matrix $\widehat{\mathbf{X}}$ filled with missing values, each of $\mathbf{X}$ which is a potential value predicted through global contextual information. However, the results generated by this step may not be precise enough in terms of local features, especially on complex and diverse datasets.

Next, we combined the KNNImputer method to fine tune and optimize the candidate features generated by the mask decoder. Specifically, for each missing value $\mathbf{X}_{ij}$, KNNImputer utilizes local neighbor information for fine-grained filling. Firstly, calculate the distance matrix $\mathbf{D}$ for each sample in the dataset, which is not only based on the observed original data features but also incorporates candidate features $\hat{X}$ generated by the mask decoder as in (11).

$$\mathbf{D}_{i,l} = \sqrt{\sum_{m\in\text{observed}}(\mathbf{X}_{im} - \mathbf{X}_{lm})^2 + \lambda \sum_{m\in\text{missing}}(\widehat{\mathbf{X}}_{im} - \widehat{\mathbf{X}}_{lm})^2}$$

..... [Formular 11]

where, λ is an adjustment parameter used to balance the influence of raw data features and candidate features. After calculating the distance matrix, select the k most similar neighbors, and then use the eigenvalues of these neighbors to perform weighted average filling on the missing data as in (12).

$$\widehat{\mathbf{X}}_{ij}^{\text{final}} = \frac{\sum_{p=1}^{k} w_p \mathbf{X}_{l_p j}}{\sum_{p=1}^{k} w_p} \text{..... [Formular 12]}$$

where, $\mathbf{X}_{l_p j}$ is the eigenvalue of the p-th neighbor in the j-th column, $w_p = 1/\mathbf{D}_{i,l_p}$ represents the reciprocal of the distance to the neighbor. Through this combination method, not only the features of local neighbors are referred to, but also the global context information provided by the mask decoder is considered, thereby achieving more accurate and consistent missing value filling

The combination of KNNImputer and mask decoder can effectively utilize global and local information to fill missing values, thereby providing better filling results on complex datasets. This methodology provides a powerful and flexible solution for handling missing data, suitable for various application scenarios.

4. Experiment

In this research, the computational environment was constructed using a high-performance GPU cluster equipped with multiple NVIDIA RTX A5000 units and Intel 14th-generation processors, enabling high-throughput visual data processing and efficient training of deep neural architectures. The system operates on Ubuntu 20.04, while PyTorch 2.4.0 serves as the primary development framework to maximize GPU acceleration. All experiments were conducted using Python 3.7.1. This configuration ensures stable large-scale training, particularly important for contrastive SSL pretraining and graph-based temporal modeling.

4.1 Dataset and Configuration

To rigorously evaluate the proposed self-supervised visual neural encoding framework, this study uses two widely recognized benchmark datasets CIFAR-10 and SVHN. These datasets are intentionally chosen because they contain diverse visual structures, noisy samples, and low-resolution images, making them ideal for testing the robustness of the proposed Mask-Aware Gating (MAG) mechanism and the contrastive encoding pipeline.

CIFAR-10 [20]

The CIFAR-10 dataset, created by the Canadian Institute for Advanced Research, contains **60,000 color images** at a resolution of 32×32 pixels. These images span **10 object categories**, including

airplane, automobile (excluding trucks and pickup trucks), bird, cat, deer, dog, frog, horse, ship, and truck. Each category includes 6,000 samples, with 5,000 allocated for training and 1,000 for testing.

CIFAR-10 is particularly suitable for validating the spatial feature extraction component of the encoding model, as its high inter-class variability and limited resolution challenge both CNN-based and transformer-based encoders. This helps evaluate whether the proposed MAG mechanism improves feature robustness under ambiguous or low-quality visual conditions.

SVHN [21]

The SVHN dataset is derived from Google Street View house numbers and contains over 600,000 labeled RGB digit images, each sized 32×32 pixels. It is divided into three parts:

- 73,257 training images,
- 26,032 testing images, and
- 531,131 additional images for extended training.

SVHN introduces real-world noise, cluttered backgrounds, illumination variations, and digit occlusions. These properties make it an ideal benchmark for evaluating how well the proposed SSL encoder especially under masked or missing data generalizes to real-world visual distortions.

Why These Datasets?

Unlike many prior visual neural encoding studies that rely solely on high-quality laboratory datasets, this paper purposefully employs CIFAR-10 and SVHN to test:

- robustness to noise and low resolution,
- effectiveness of SSL without large annotated datasets, and
- adaptability of MAG under occlusion and missing information.

These challenges mirror conditions encountered in real-world visual perception and thus better validate the general applicability of the proposed encoding model.

ImageNet [22]: The ImageNet dataset contains 14,197,122 annotated images constructed based on the WordNet hierarchy, and has long been considered a challenging dataset in the fields of image classification and object detection. This dataset publicly provides a set of manually annotated training images, as well as some test images, but their manual annotations are retained. ILSVRC annotation is mainly divided into two types: image level annotation and object level annotation.

By default, the model architecture uses a two-layer MLP as the projection head, with an output dimension set to 256. The optimizer uses AdamW, and the initial learning rate is adjusted according to the batch size to $5 \times 10^{-4} \times \frac{Batch\ Size}{256}$, and combined with a weight attenuation coefficient of 0.05.

The learning rate scheduling adopts a strategy combining linear preheating and cosine annealing, with a linear increase in the first 10 epochs and a gradual decay following cosine annealing. During the training process, the number of epochs for CIFAR-10, SVHN, and ImageNet were set to 500, 300, and 600, respectively, with a batch size of 1024. Pre training was conducted on ImageNet.

4.2 Evaluating Indicator

In self supervised learning, the evaluation metrics of the model are different from traditional supervised learning, as self supervised tasks mainly focus on the quality of feature representations learned by the model from unlabeled data. To measure the effectiveness of these feature representations, a series of evaluation metrics based on downstream tasks and feature distributions are typically used.

Linear Evaluation: Train a simple linear classifier on top of the fixed features extracted by the self supervised model as input. Assuming that the features generated by the self supervised model are $\mathbf{Z} = f(\mathbf{X}; \theta)$, where x is the input image, $\mathbf{X}$ is the model parameter, $\mathbf{Z}$ is a feature representation. Linear detectors are trained by minimizing the following cross entropy loss as in (13).

$$\mathcal{L}_{\text{linear}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} y_{i,c} \log(\sigma(\mathbf{W}^{\top}\mathbf{Z}_i)_c) \cdots \text{[Formular 13]}$$

Among them, N is the number of samples, C is the number of categories, $y_{i,c}$ is the true label of sample i

Top-1 and Top-5 evaluation indicators: Top-1 accuracy refers to whether the predicted category of the model is completely consistent with the true category of the sample. In the classification task, for each input image, the model generates probability distributions for all possible categories and selects the category with the highest probability as the prediction result. The Top-1 accuracy calculation formula is as follows as in (14).

$$Top-1\ Accuracy = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(\hat{y}_i = y_i) \cdots\cdots\cdots \text{[Formular 14]}$$

$1(\cdot)$ is an indicator function, which takes the value of 1 when the predicted category matches the true category, and 0 otherwise. Top-1 accuracy measures the model's best predictive ability for each sample, that is, whether the model can correctly identify the category of the sample in the first option.

Top-5 accuracy considers whether the top five categories with the highest prediction probability of the model contain true categories. That is to say, if the true category is among the top five most probable predicted categories given by the model, then this prediction is considered correct. The Top-5 accuracy calculation formula as in (15).

$$Top-5\ Accuracy = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}(y_i \in \{\hat{y}_i^1, \hat{y}_i^2, \hat{y}_i^3, \hat{y}_i^4, \hat{y}_i^5\}) \text{[Formular 15]}$$

$\hat{y}_i^1, \ldots, \hat{y}_i^5$ are the five categories with the highest prediction probability for the i-th sample by the model. Top-5 accuracy provides more fault-tolerant space for the model, making it suitable for scenarios with a large number of categories and small differences between categories.

4.3 Self supervised tasks on CIFAR-10 and SVHN

Table 1. Comparison of methods on CIFAR-10 and SVHN datasets.

Method	Learning Way	CIFAR-10 (4K) Mean ± std (%)	SVHN-1K Mean ± std (%)
Mean Teacher [23]	Label Propagation	84.1 ± 0.3	94.4 ± 0.5
VAT + EntMin [24]		86.8 ± 0.4	94.6 ± 0.2
LGA + VAT [25]		87.5 ± 0.5	–
ICT [26]		92.7 ± 0.1	–
EnAET [27]		92.9 ± 0.1	–
MixMatch [28]		93.7 ± 0.1	96.7 ± 0.3
UDA [29]		94.5 ± 0.2	97.1 ± 0.2
FixMatch [30]		95.7 ± 0.1	97.1 ± 0.4
S-MAG (Ours)		95.1 ± 0.3	98.2 ± 0.3
BYOL [13]	Self Supervised Training	–	68.8
MOCOv2 [31]		–	71.1
SimCLR [19]		–	71.7
PIRL [4]		–	84.9
PCL [31]		–	85.6
S-MAG (Ours)		92.7 ± 0.5	93.9 ± 0.1

Table 1 shows the performance of S-MAG with several advanced supervised learning methods (such as UDA, FixMatch, etc.) and self supervised learning methods (such as SimCLR, PCL, etc.) on the CIFAR-10 (4K) and SVHN-1K datasets. In supervised learning, FixMatch performs the best on CIFAR-10 (4K) with an average accuracy of 95.7%, while UDA achieves an average accuracy of 97.1% on the SVHN-1K dataset. S-MAG (the method proposed in this article) also performs well under supervised learning, with accuracies of 95.1% and 98.2% on CIFAR-10 (4K) and SVHN-1K, respectively. In contrast, the performance of self supervised learning methods is more diverse, with PCL and PIRL performing better on SVHN-1K with accuracies of 85.6% and 84.9%, respectively, while SimCLR and BYOL have relatively lower accuracies of 71.7% and 68.8%, respectively. S-MAG performs equally well under self supervised learning, with accuracies of 92.7% and 93.9% on CIFAR-10 (4K) and SVHN-1K, respectively. Overall, S-MAG has demonstrated excellent performance in both supervised and self supervised learning, particularly in terms of strong generalization ability on high complexity datasets.

4.4 Fine Tuning and Label Transfer on ImageNet

Table 2. The Top-1 and Top-5 classification accuracy of different methods under SSL pre training and no pre training conditions (for the 1% and 10% ImageNet datasets), respectively.

Method	SSL Pretrain Method	Training Epoch	Top-1 (%)		Top-5 (%)	
			1%	10%	1%	10%
UDA[29]	w/o Pre	–	–	67.4	–	87.9
VAT+EntMin[33]		–	–	68.8	–	88.5
Pseudo-label[33]		100	–	–	51.6	82.4
CoMatch[34]		400	66.0	73.6	86.4	91.6
Fine-tune	PCL	200	–	–	75.3	85.6
	SimCLR V2	800	57.9	68.4	82.5	89.2
	BYOL	1000	53.2	68.8	78.4	89.0
	SwAV	800	53.9	70.2	78.5	89.9
	WCL	800	65.0	72.0	86.3	91.2
Fine-tune	MoCo V2	800	49.8	66.1	77.2	87.9
CoMatch[34]		1200	67.1	73.7	87.1	91.4
Pseudo-label	S-	600	65.9	74.5	89.6	90.7
Fine-tune	MAG(Ours)	600	69.7	77.4	90.2	92.5

Table 2 shows the performance of several self supervised learning (SSL), pre training methods on different data subset ratios (1% and 10%), as well as their Top-1 and Top-5 classification accuracy after fine-tuning using the ImageNet dataset. The main contents of the table include method names, SSL pre training methods, training epochs, and Top-1 and Top-5 classification accuracies under different dataset ratios. Different self supervised learning methods perform differently on different proportions of datasets. For example, although UDA and VAT+EntMin were not pre trained (w/o Pre.), they achieved Top-1 accuracy of 67.4% and 68.8%, as well as Top-5 accuracy of 87.9% and 88.5%, respectively, at a dataset ratio of 10%. This indicates that although these methods have not undergone self supervised pre training, they can still achieve relatively high classification accuracy with less labeled data. It is worth noting that after 200 training rounds, PCL achieved a Top-5 accuracy of 75.3% on a 1% dataset, while SimCLR V2 achieved Top-1 accuracy of 57.9% and 68.4% on dataset after 800 training rounds, and Top-5 accuracy of 82.5% and 89.2%, respectively. This indicates that SimCLR V2 still cannot surpass PCL's performance over a longer training period. BYOL and SwAV showed similar Top-1 accuracy at 1% and 10% dataset ratios after 1000 and 800 training rounds, respectively, with accuracies of 53.2% and 68.8%, as well as 53.9% and 70.2%. On the combination of CoMatch and MoCo V2, we observed that after 800 and 1200 training epochs, the Top-1 accuracies were 49.8% and 67.1%, respectively, on a 1% dataset, while the Top-1 accuracies were 66.1% and 73.7%, respectively, on a 10% dataset. In terms of Top-5 accuracy, these two methods perform relatively stably under different training epochs, with rates of 77.2% and 87.1%, as well as 87.9% and 91.4%, respectively.

Finally, S-MAG (the method proposed in this article) underwent 600 training rounds at 1% and

10% dataset ratios, respectively. Without performing Pseudo label fine-tuning, the Top-1 accuracy of S-MAG was 65.9% and 74.5% at 1% and 10% dataset ratios, respectively, while the Top-5 accuracy was 89.6% and 90.7%. After fine tuning, the Top-1 accuracy of S-MAG significantly improved, reaching 69.7% and 77.4% respectively, and the Top-5 accuracy also reached 90.2% and 92.5%. This result indicates that S-MAG, after self supervised pre training and fine-tuning, can significantly improve classification accuracy with less labeled data, and its performance is superior to other methods listed in the table.

4.5 Ablation Study

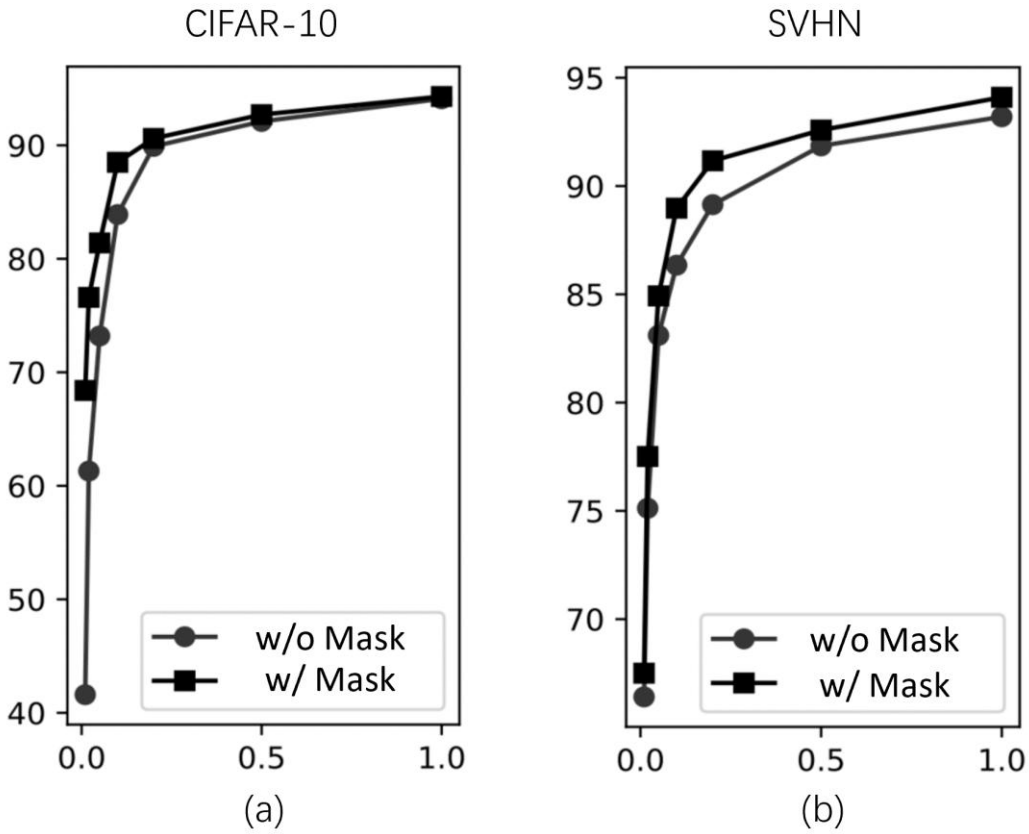

Figure 3. The impact of w/ and w/o mask strategies on model performance.

The vertical axis represents classification accuracy, and the horizontal axis represents training data epoch ratios.

The impact of occlusion strategy on pre training: We demonstrate the impact of using and not using masking techniques on model performance on the CIFAR-10 and SVHN datasets in Figure 3, and present the visualization of random masks for different images in Figure 4. The results indicate that models using masks exhibit higher accuracy during training, especially in the early stages of training, in both CIFAR-10 and SVHN datasets. Masking techniques significantly improve the learning speed and performance of the models. As the training progressed, the performance of both methods gradually stabilized, but the model using masks still maintained a slightly higher accuracy in the end. In the CIFAR-10 dataset, the accuracy of models using masks is close to 93%, which is

about 3% higher than models without masks. In the SVHN dataset, the model using masks achieved an accuracy of approximately 95%, which is slightly higher than the model without masks. This indicates that masking technology plays a significant role in improving the early learning efficiency and final performance of the model.

Figure 4. Random mask visualization (token Size is 20×20, token number is 200).

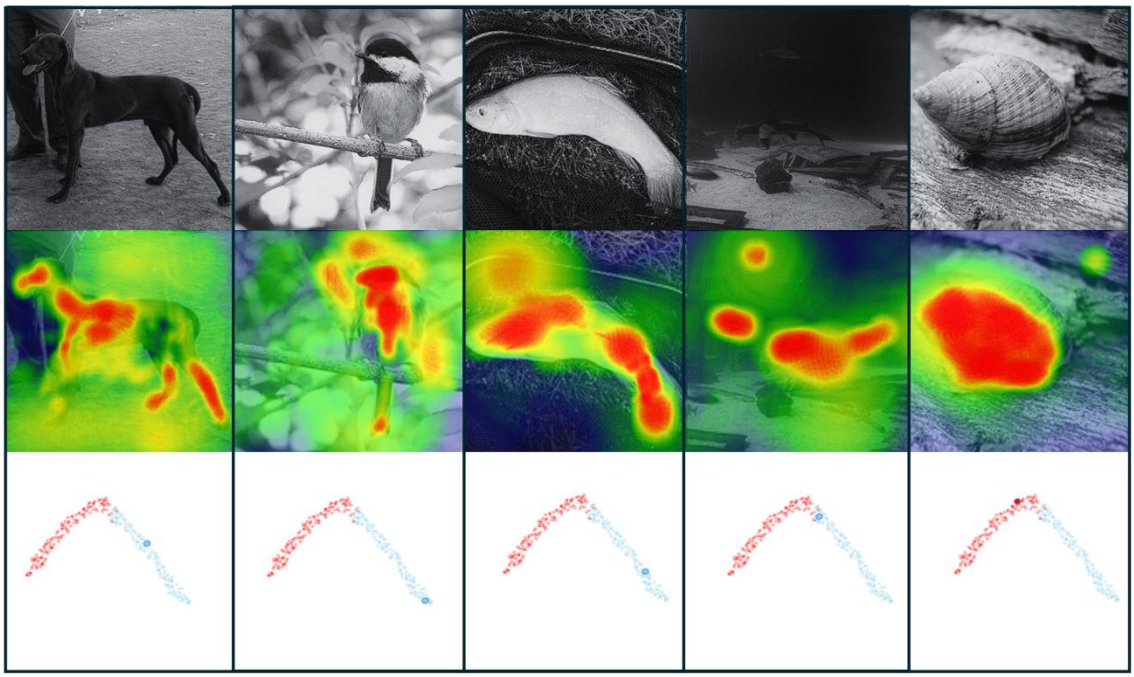

Figure 5. Visualization of gated attention mechanism.

Table 3. Test with different token sizes and numbers.

Token Size	Token Number	Pre. Model	ImageNet (1%)	
			Top-1	Top-5
10×10	100	ResNet-50	58.2	80.9

10×10	200	S-MAG	64.5	86.1
		ResNet-50	66.4	87.5
		S-MAG	68.9	86.4
20×20	100	ResNet-50	62.6	81.5
		S-MAG	65.8	88.9
20×20	200	ResNet-50	67.4	87.0
		S-MAG	69.7	90.2

Table 3 shows the classification performance of ResNet-50 and S-MAG pre trained models on the ImageNet 1% dataset under different token sizes (10x10 and 20x20 pixels) and token quantities (100 and 200). The results indicate that as the number of tokens increases, the accuracy of the Top-1 and Top-5 models significantly improves, which has a significant impact on the results. When the token size is large and the quantity is small, the enhancement effect is low. Larger occlusion blocks may cover key parts of the image, especially as the number of occlusion blocks increases, the effective information of the image may be covered to a larger proportion. The effect may not be significant when the Token Number is small, but more useful features can be learned better when the Token Number is large.

The impact of gated attention: Table 4 shows that the introduction of Self Attention Module (SAM) and Gated Attention Module (GAM) significantly improved the performance of the model on the CIFAR-10 and ImageNet datasets. The benchmark model (without attention module) has a Top-1 accuracy of 90.1% on CIFAR-10, and Top-1 and Top-5 accuracies of 55.3% and 73.8% on ImageNet, respectively, with a computational cost of 2.6G FLOPs. After adding SAM, the Top-1 accuracy of CIFAR-10 increased to 92.8%, while the Top-1 and Top-5 accuracies of ImageNet increased to 64.5% and 88.1%, respectively. However, the computational cost also significantly increased to 15.3G FLOPs. When using GAM, the Top-1 accuracy of CIFAR-10 was further improved to 95.1%, while the Top-1 and Top-5 accuracies of ImageNet reached 69.7% and 90.2%, respectively. At the same time, the computational cost increased to 17.5G FLOPs. Overall, GAM performs the best in improving model performance, but its high computational cost needs to be balanced in practical applications.

Table 4. The impact of attention strategy and comparison with FLOP.

Model	CIFAR-10 (LP)	ImageNet-1% (SSL)		FLOPs
		Top-1	Top-5	
Baseline	90.1	55.3	73.8	2.6G
Baseline + SAM	92.8	64.5	88.1	15.3G
Baseline + GAM	95.1	69.7	90.2	17.5G

In Figure 5, we present the visualization of some images. The first raw shows the preprocessed image, the second raw represents the attention of GAM to different samples, and the third raw represents the predicted category. The blue dotted area represents the predicted positive distribution,

the red dotted area represents the predicted negative distribution, and the fifth image is an example of incorrect prediction.

5. Conclusion

This article proposes an innovative Mask-Aware Gating (MAG) mechanism for self-supervised visual neural encoding. By introducing learnable mask structures and mask-guided interpolation strategies, the model significantly enhances its ability to extract stable and informative features under challenging visual conditions. Compared with traditional SSL-based encoders, the proposed method achieves superior performance across multiple benchmark datasets (particularly when processing is partially occluded, corrupted) or noisy images. These results confirm that MAG effectively addresses two key limitations in existing visual encoding frameworks: insufficient robustness to incomplete inputs and limited adaptability to real-world visual distortions.

The main contributions of this study include the development of:

- a mask-aware gating module for adaptive attention regulation,
- a visual masking and interpolation framework that simulates realistic data-loss scenarios, and
- a comprehensive evaluation pipeline validating the model under both supervised and self-supervised tasks.

Together, these contributions establish a coherent and biologically inspired encoding model capable of learning resilient visual representations without requiring large annotated datasets. The findings further demonstrate that integrating mask-awareness into SSL architectures provides a powerful mechanism for improving generalization, computational efficiency, and noise tolerance.

Beyond theoretical significance, MAG shows strong potential for deployment in practical applications such as robotics, surveillance, medical imaging, autonomous driving, and sports movement analysis -- domains in which visual data are often incomplete, occluded, or noisy.

Looking ahead, several research directions may further enhance the effectiveness of the proposed approach:

1. Optimizing the computational complexity of MAG to support large-scale, real-time visual tasks.
2. Extending MAG to multimodal or high-resolution vision systems to examine its scalability and versatility.
3. Integrating MAG with emerging deep learning paradigms (e.g., vision transformers, sparse attention, diffusion models) to improve representation diversity and interpretability.
4. Exploring downstream temporal tasks such as pose estimation and action recognition to further demonstrate the robustness of MAG-enhanced neural encoding in dynamic environments.

Overall, this work establishes Mask-Aware Gating as a promising direction for building stronger and more resilient self-supervised visual encoding models, laying the foundation for next-generation intelligent visual systems.

Acknowledgements

This article received no financial or funding support.

Conflicts of Interest

The author confirms that there are no conflicts of interest.

References

[1] He, K., Zhang, X., Ren, S. and Sun, J. Deep residual learning for image recognition In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, 770–778.

[2] Ning, X., Tian, W., Yu, Z., Zhang, Y., He, F. and Li, Y. HCFNN: high-order coverage function neural network for image classification Pattern Recognition, 2022, 131, 108873.

[3] Guger, C., Ince, N.F., Korostenskaja, M., Allison, B.Z., Daly, I. and Müller-Putz, G. Brain-computer interface research: a state-of-the-art summary Brain-Computer Interface Research: A State-of-the-Art Summary 11 Berlin: Springer, 2024, 1–11.

[4] Misra, I. and van der Maaten, L. Self-supervised learning of pretext-invariant representations In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 6707–6717.

[5] Jing, L. and Tian, Y. Self-supervised visual feature learning with deep neural networks: a survey IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 43(11), 4037–4058.

[6] Yang, K., Cao, Y., Zhang, Y., Wang, Y. and Liu, J. Self-supervised learning and prediction of microstructure evolution with convolutional recurrent neural networks Patterns, 2021, 2(5).

[7] Chen, X., Ding, M., Wang, X., Li, Z. and Tao, D. Context autoencoder for self-supervised representation learning International Journal of Computer Vision, 2024, 132(1), 208–223.

[8] Kang, B., Lee, W., Seo, H., Kim, S. and Park, H.J. Self-supervised learning for denoising of multidimensional MRI data Magnetic Resonance in Medicine, 2024, 92(5), 1980–1994.

[9] Liu, Z. and Han, X.H. Hyperspectral image super resolution using deep internal and self-supervised learning CAAI Transactions on Intelligence Technology, 2024, 9(1), 128–141.

[10] Kumar, P., Rawat, P. and Chauhan, S. Contrastive self-supervised learning: review, progress, challenges and future research directions International Journal of Multimedia Information Retrieval, 2022, 11(4), 461–488.

[11] Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. A simple framework for contrastive learning of visual representations In: International Conference on Machine Learning, 2020, 1597–1607.

[12] He, K., Fan, H., Wu, Y., Xie, S. and Girshick, R. Momentum contrast for unsupervised visual representation learning In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, 9729–9738.

[13] Grill, J.B., Strub, F., Altché, F., Tallec, C., Richemond, P.H., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Azar, M.G., Piot, B., Kavukcuoglu, K., Munos, R. and Valko, M. Bootstrap your own latent: a new approach to self-supervised learning Advances in Neural Information Processing Systems, 2020, 33, 21271–21284.

[14] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. Generative adversarial networks Communications of the ACM, 2020, 63(11), 139–144.

[15] Zou, Y., Liao, S. and Wang, Q. Chinese image captioning with fusion encoder and visual keyword search IET Image Processing, 2024.

[16] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M.,

Heigold, G., Gelly, S., Uszkoreit, J. and Houlsby, N. An image is worth 16×16 words: transformers for image recognition at scale arXiv preprint arXiv:2010.11929, 2020.

[17] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. and Guo, B. Swin transformer: hierarchical vision transformer using shifted windows In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 10012–10022.

[18] Ghorvei, M., Kavianpour, M., Beheshti, M.T., Hosseini, R. and Ghorvei, H. Synthetic to real framework based on convolutional multi-head attention and hybrid domain alignment In: 2022 8th International Conference on Control, Instrumentation and Automation, 2022, 1–6.

[19] Chen, T., Kornblith, S., Swersky, K., Norouzi, M. and Hinton, G. Big self-supervised models are strong semi-supervised learners Advances in Neural Information Processing Systems, 2020, 33, 22243–22255.

[20] Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images, 2009.

[21] Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B. and Ng, A.Y. Reading digits in natural images with unsupervised feature learning In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, 2011, 1–4.

[22] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K. and Fei-Fei, L. ImageNet: a large-scale hierarchical image database In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, 248–255.

[23] Tarvainen, A. and Valpola, H. Mean teachers are better role models: weight-averaged consistency targets improve semi-supervised deep learning results Advances in Neural Information Processing Systems, 2017, 30.

[24] Miyato, T., Maeda, S.I., Koyama, M., Nakae, K. and Ishii, S. Virtual adversarial training: a regularization method for supervised and semi-supervised learning IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018, 41(8), 1979–1993.

[25] Jackson, J. and Schulman, J. Semi-supervised learning by label gradient alignment arXiv preprint arXiv:1902.02336, 2019.

[26] Verma, V., Kawaguchi, K., Lamb, A., Kannala, J., Bengio, Y. and Lopez-Paz, D. Interpolation consistency training for semi-supervised learning Neural Networks, 2022, 145, 90–106.

[27] Wang, X., Kihara, D., Luo, J. and Qi, G.J. ENAET: a self-trained framework for semi-supervised and supervised learning with ensemble transformations IEEE Transactions on Image Processing, 2020, 30, 1639–1647.

[28] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A. and Raffel, C. MixMatch: a holistic approach to semi-supervised learning Advances in Neural Information Processing Systems, 2019, 32.

[29] Xie, Q., Dai, Z., Hovy, E., Luong, M.T. and Le, Q.V. Unsupervised data augmentation for consistency training Advances in Neural Information Processing Systems, 2020, 33, 6256–6268.

[30] Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C., Cubuk, E.D., Kurakin, A. and Li, C.L. FixMatch: simplifying semi-supervised learning with consistency and confidence Advances in Neural Information Processing Systems, 2020, 33, 596–608.

[31] Chen, X., Fan, H., Girshick, R. and He, K. Improved baselines with momentum contrastive learning arXiv preprint arXiv:2003.04297, 2020.

[32] Li, J., Zhou, P., Xiong, C., Socher, R. and Hoi, S.C.H. Prototypical contrastive learning of unsupervised representations arXiv preprint arXiv:2005.04966, 2020.

[33] Zhai, X., Oliver, A., Kolesnikov, A. and Beyer, L. S4L: self-supervised semi-supervised learning In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, 1476–1485.

[34] Li, J., Xiong, C. and Hoi, S.C.H. CoMatch: semi-supervised learning with contrastive graph regularization In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 9475–9484.

[35] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P. and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments Advances in Neural Information Processing Systems, 2020, 33, 9912–9924.

[36] Zheng, M., Wang, F., You, S., Qian, X. and Zhang, C. Weakly supervised contrastive learning In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, 10042–10051.

[37] Wu, S., Sun, F., Zhang, W., Xie, X. and Cui, B.. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 2022, *55*(5), 1-37.

[38] Ma, Y., Li, H. and Yan, H. Efficient Real-Time Sports Action Pose Estimation via EfficientPose and Temporal Graph Convolution. *IEEE Access*, 2025.