Available online at **www.joac.info**

ISSN: 2278-1862

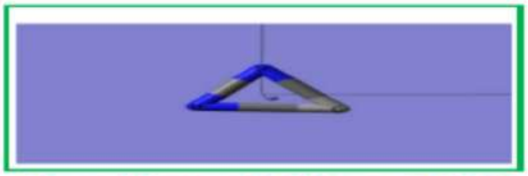
Journal of Applicable Chemistry

2025, 14 (2): 279-394

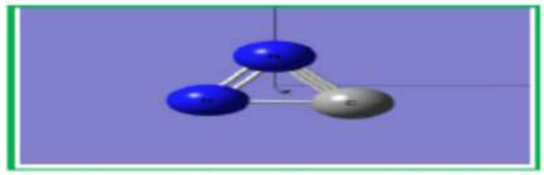
(International Peer Reviewed Journal)

New Chemistry News

$^{-}N=C=N^{-}$



New News of Chem (NNC)



ChemNewsNew (CNN)

CNN – 66B

Iam (Intelligence Augmented /Assisted Method(s))

Transformers --architectures & Fits (2025)

Information Source	**sciencedirect.com;**	
S. Narasinga Rao M D *Associate Professor,* *Emergency Medicine dept.,* *Andhra Medical College,* *King George Hospital* *Visakhapatnam, A.P., India*	*K. Somasekhara Rao, Ph D* *Dept. of Chemistry,* *Acharya Nagarjuna Univ.,* *Dr. M.R.Appa Rao Campus,* *Nuzvid-521 201, India*	*R. Sambasiva Rao, Ph D* *Dept. of Chemistry,* *Andhra University,* *Visakhapatnam 530 003,* *India*
snrnaveen007@gmail.com (+91 98 48 13 67 04)	sr_kaza1947@yahoo.com (+91 98 48 94 26 18)	rsr.chem@gmail.com (+91 99 85 86 01 82)

Conspectus: A Transformer net (TransF Net) or Transformer neural network (TransF NN) consists of an attention layer and MLP-NN to carry out Natural Language processing (NLP). The seminal paper of Ashish Vaswani et al. entitled "Attention is All You Need", in the year 2017 revolutionised sequence text data processing. The evolution of architecture of TransF NN, attention mechanism, and hybridization with other methods, during these few years, brought this paradigm to the fore-front of Data

Science dealing with Text, numerical time-series, sound (speech), images and video sequence with local and global inter-dependencies.

Data: Data are many types, viz., Boolean (AND (Conjunction), OR (Disjunction), NOT(Negation) …), numerical (integer, floating point), sound (speech), text, image, video, and that from tactile senses (touch, smell, taste etc.). Real and complex (and quaternion, octonion, …) belong to real and imaginary spaces. Depending upon bases of items, tensors (zero, first, second, third order or scalar, vector, matrix, tensor-of-third-order) are popular. Higher (>3) order tensors are also sometimes called as multi (4-, 5-,6-) –way-data. Data are also referred as binary, octal, decimal and hexa-decimal with base 2,8,10 and 16 respectively. In images pixels for two-dimensional and voxels for 3-dimensional spaces are standard with black/white, RGB, and grey-colors. An image is also conceived as a culmination of patches.

Text data contain documents, pages, paragraphs, sentences, words and alphabetic characters. Each datum for example, a word in text, amino-acid in protein, real numerical value in stock market/forex field (time series), or patch in an image is called a token. The position of token in the sequence and numerical (embedded) value are the primary input for analysis. A few typical tasks in NLP are translation of text in one language (English) to another (German or French or …), summarizing documents, generation of new stories in text format and so on.

The architectures of a few of Transformer neural nets (TransF-NN) documented in this state-of-knowledge-methods-module for Data2Knowledge transformation are

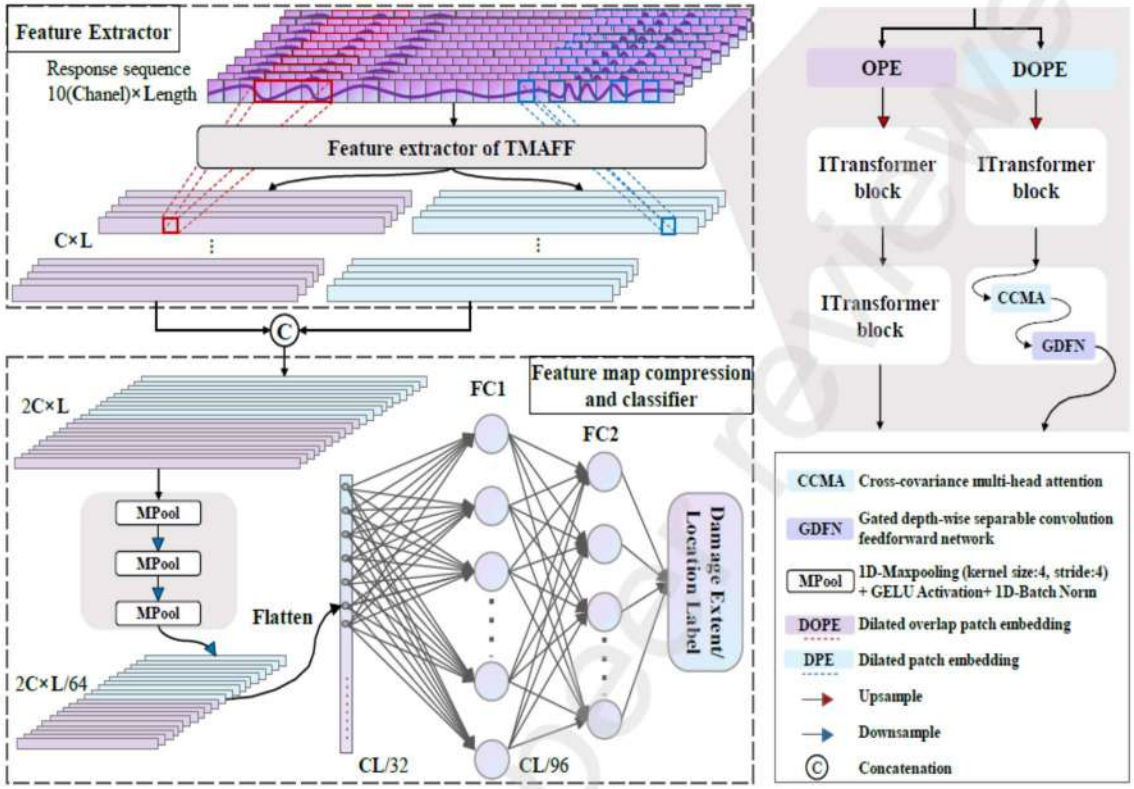
- Multi-scale Acceleration Feature Fusion (MAFF) Transformer (TMAFF)
- Video second-order transformer network (ViSoT)
- DHCT-GAN Transformer (DHCT: Dual-branch Hybrid CNN–Transformer Network, GAN: generative adversarial network)
- Discrete Cosine Transform Network (DCTNet)
- Hierarchical Cross-Modal Similarity Search and Transfer (HCMSST)
- Enhanced Hybrid of CNN and Transformer Network (EHCTNet)
- Swin Transformer (SwinTrnF)
- Spatial-Temporal Transformer Network (ST-TNet)
- Hybrid Attention-Dense Connected Transformer (HADT) Networks
- Efficient Conv-Transformer (ECTFormer)
- Model architecture of MultiscAle Relational Transformer network (MART)
- Multifractal Spectrum Transformer (MFSTransFormer)
- Residual sparse Transformer branch (RSTB) ;{[MSFN Mixed scale FFNN Top C sparse attention (TCSA)]}
- Parallel pyramidal Transformer [MSD (Multi-Scale Decoder), PPT, (Parallel Processing Transformer) FM. (Feature Modulation)]
- SCACD-Net [SCA (Spatial and Channel Attention) CD (Contextual Decoder)]
- Digital image correlation (DIC) transformer (DICTr)
- TC (tea chrysanthemums) ViT
- SF-Hformer (Spatial-Frequency Hybrid Transformer)
- Multi-Patch De-raindrop Transformer for UAV images (MPDT) [contains frequency attention Transformer block (FATB) and Adaptive Feature Enhancement Module (AFEM)]

- Swin Transformer-CNN Gait (STCG) Recognition
- Histoformer [Dynamic-range Histogram Self-Attention (DHSA) module and the Dual-scale Gated Feed-Forward]
- DTCN-EFFN-Transformer [DTCN: Deep time (Temporal) convolution network structure diagram; EFFN (Feed-Forward Networks based on Expansion factor) adds an expansion factor to the linear layer]
- MDTNet (Multi-Domain Transformer Network)
- Enhanced Hierarchical Vision Transformer (EHVT); local geometric transformer; global semantic transformer
- Progressive alignment and interwoven composition network (PAIC-Net)
- Denoising Adaptive Graph Transformer (DAGT)
- Conformer (Convolution-augmented and confidence estimation network Transformer)
- MEMAFormer
- SAPPM
- Multi-Scale Convolutional Prediction Network; Multi-Scale Convolutional PoolFormer block (MSCPNet)
- MUltifuSion (MUST) Transformer
- Enhanced Downsampling Attention Block (EDAB)
- Frequency Prompting image restoration (FPro) Method

Keywords: Artificial intelligence (AI); Capsule Neural Nets— MLP-Attention Mechanism-TransFormer Nets—Hybrid TransFormer Networks--

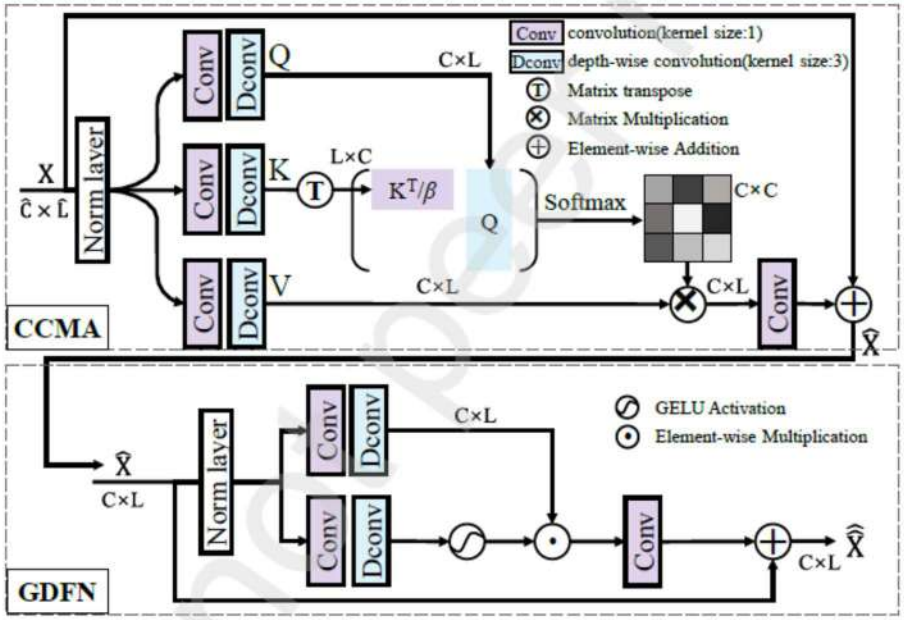
CNN : [C [Computations; Computer; Chemistry, Cell, Cellestial, Cerebrum] NN [New News; News New; Neural Nets; Nature News; News of Nature;]]
Fits : [Figure Image Table Script;]

K(nowledge)Lab
rsr.chem1979



Flowchart of

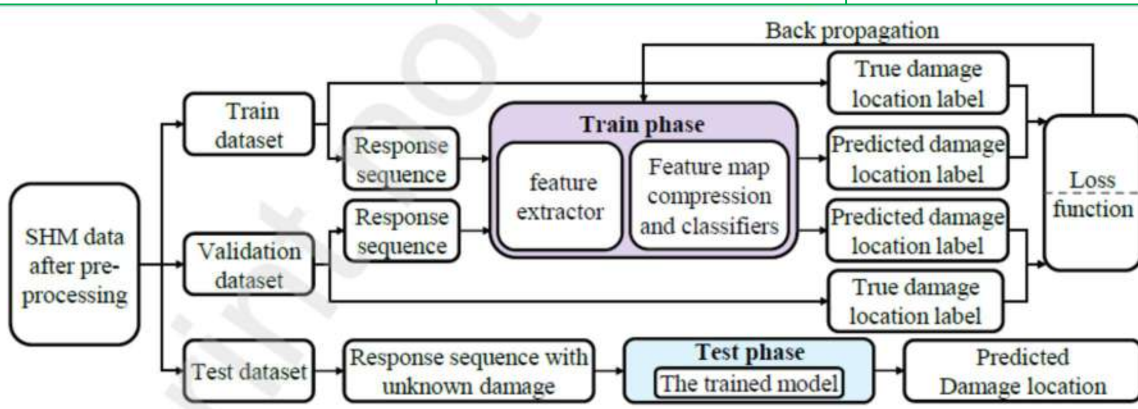
TMAFF: Transformer neural network (TNN) with Multi-scale Acceleration Feature Fusion (MAFF)



ITransformer
(Improved Transformer)

Table 1 Detailed configuration of the feature map compression and classifier

Layer	Type	Stride	Kernel size	Padding	With BN	With dropout	Activation
1	Pooling	4	4	Valid	True	P=0.1	GELU
2	Pooling	4	4	Valid	True	P=0.1	GELU
3	Pooling	4	4	Valid	True	P=0.1	GELU
4	Flatten	None	None	None	False	None	None
5	FC	None	None	None	False	P=0.1	GELU
6	FC	None	None	None	False	None	None
7	Softmax	None	None	None	False	None	None



Workflow of SDD (**Structural damage detection)** based on TMAFF

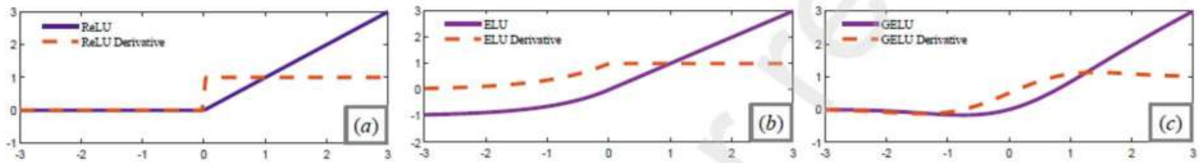
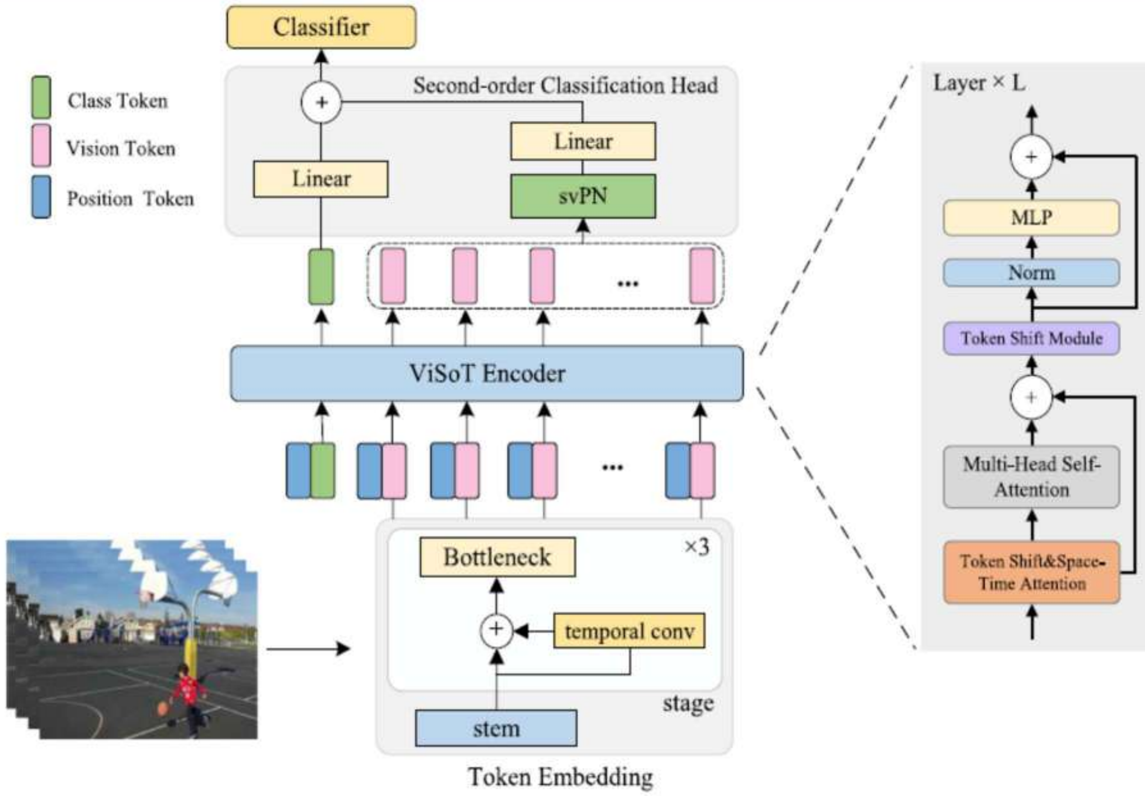


Figure 6 Activation function and derivative. (*a*) ReLU; (*b*) ELU; (*c*) GELU

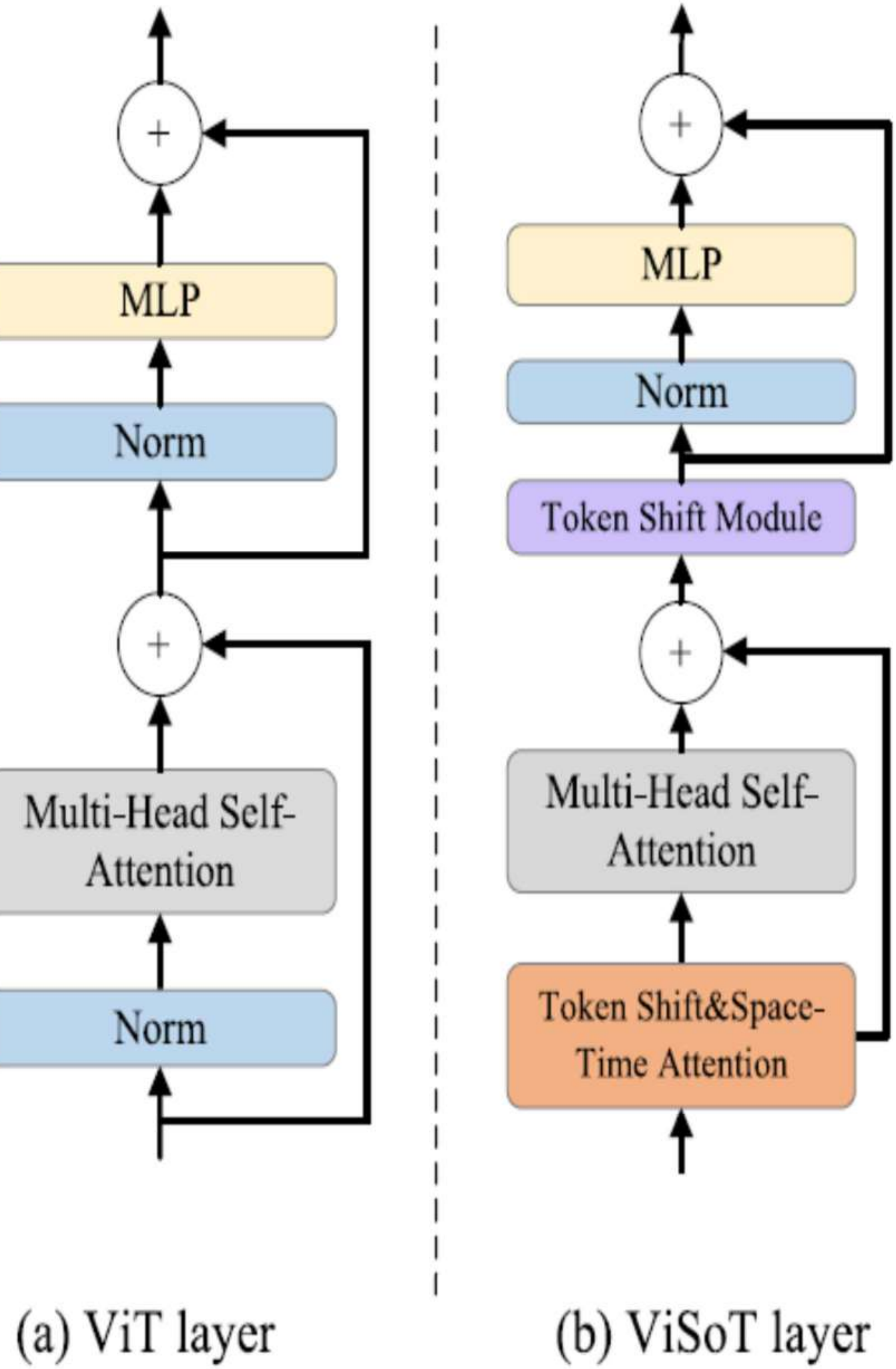
Table 4 Configurations of training process

Name	Value	Description
Batch size	256	The size of the data batch used in training
Initial learning rate	5×10^{-4}	The initial learning rate for the optimization algorithm
L2 rate	1×10^{-3}	The weight of L2 regularization
Random seed	42	The seed value used for random number generation
Patience	30	A parameter of early stopping, the acceptable duration if the performance does not increase
Total epochs	200	The total number of epochs through the entire dataset during the training process.



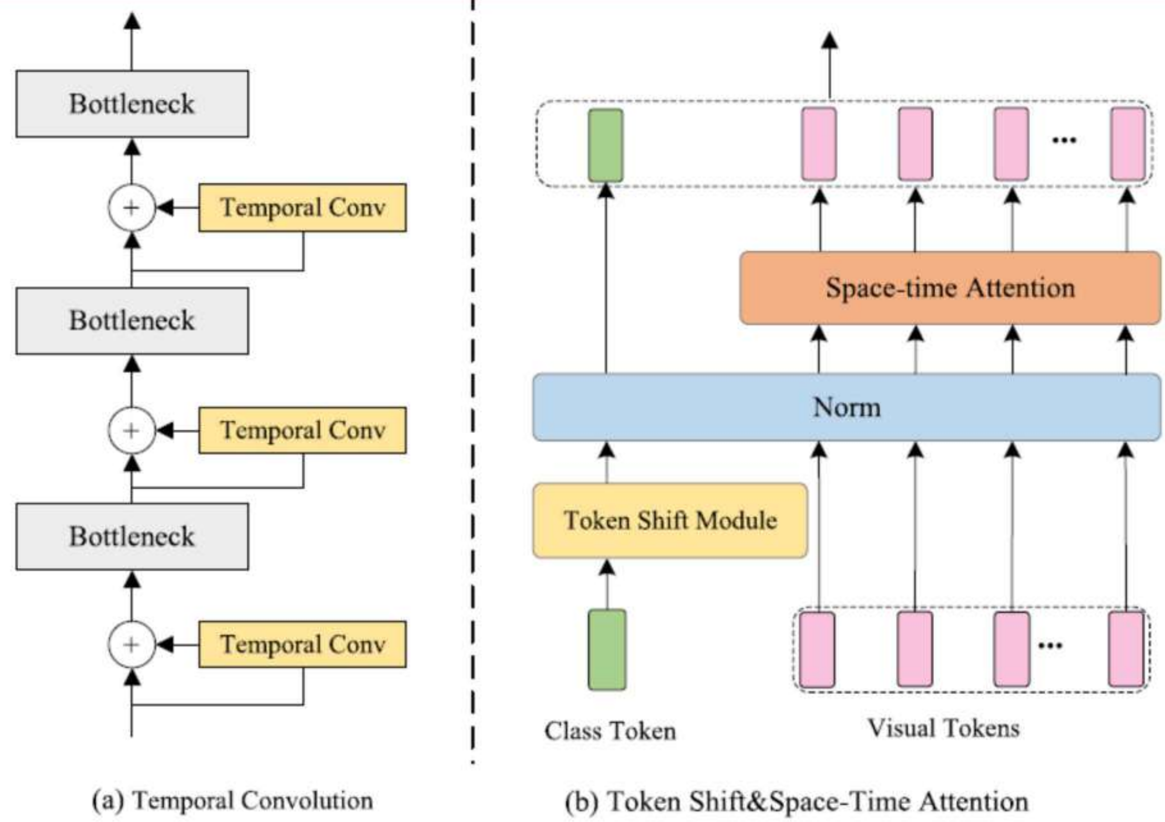
Architecture. video second-order transformer network (ViSoT)

- ViSoT: It consists of token embedding, ViSoT encoder and second-order classification head.
- The temporal convolutions are added in the stage of token embedding for temporal modeling from the start.
- Token shift module and space–time attention as in-placed components are inserted into ViSoT layers to model temporal relations and spatio-temporal interactions respectively.
- The class tokens combined with second-order aggregated visual tokens are used for classification



Comparison between standard transformer layer in ViT and present **temporal-aware ViSoT layer**.

Transformer Net	2025-02	

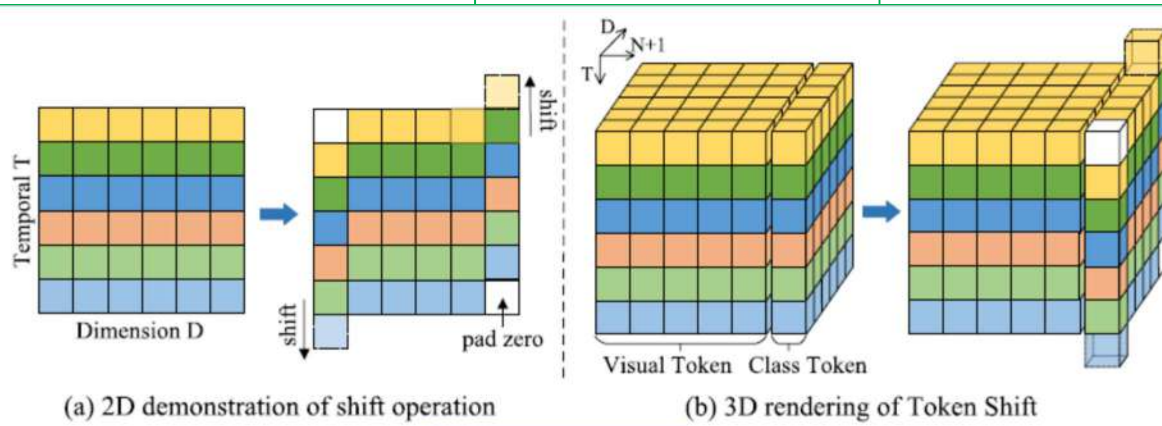


(a) Temporal Convolution (b) Token Shift&Space-Time Attention

Two temporal modeling schemes

- ✓ (a) Temporal convolution in stage of token embedding.
- ✓ (b) Token Shift & Space–Time Attention block (TS_STA) in ViSoT layer.

Transformer Net	2025-01	



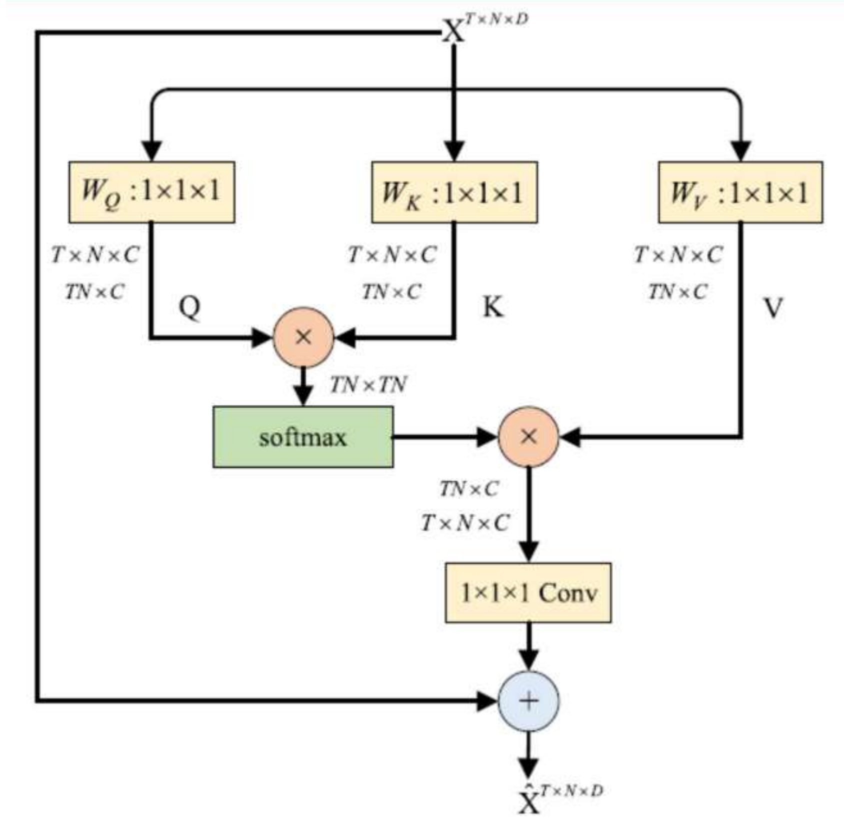
(a) 2D demonstration of shift operation (b) 3D rendering of Token Shift

Token shift operation.

- ✓ Only class token is shifted along the temporal axis
- ✓ (a) intuitively shows this operation in 2D space of class token matrix for simplicity that discards N visual tokens.

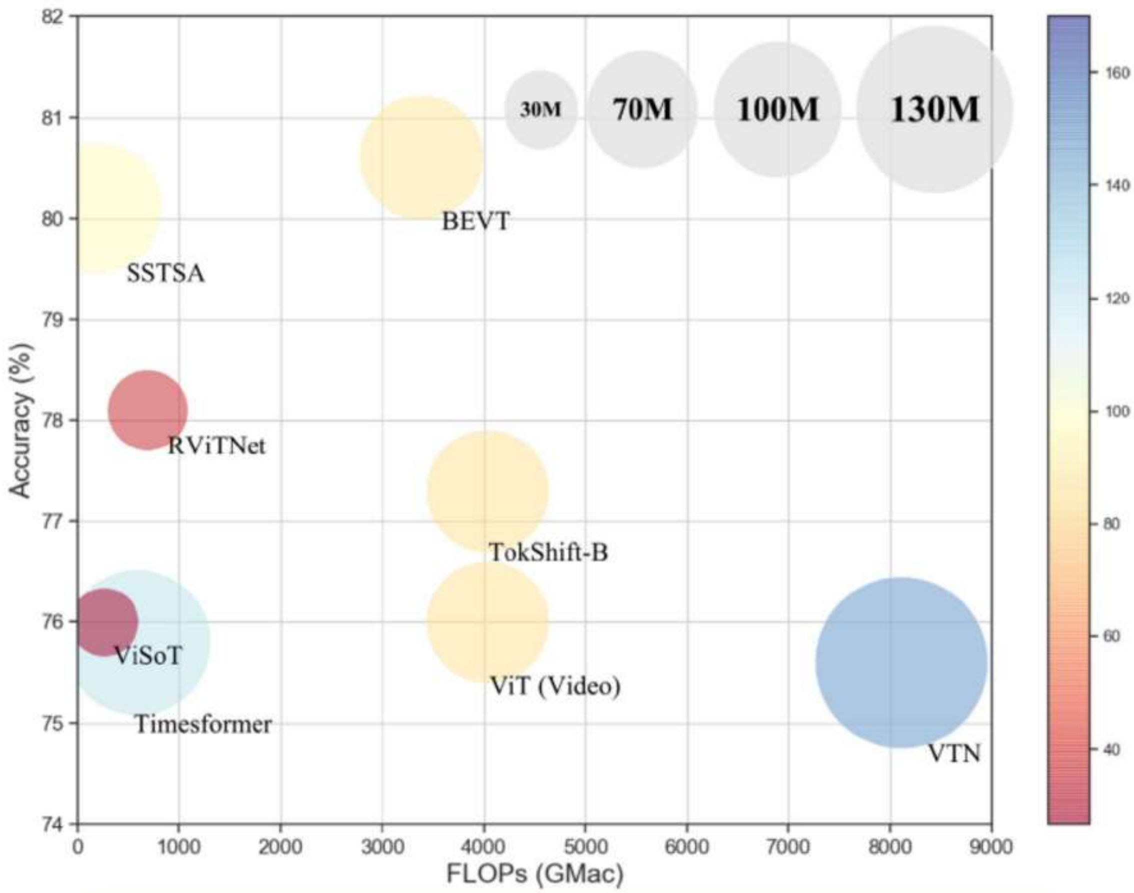
- ✓ (b) illustrates such operation from a 3D perspective of one video clip. White rectangle or cube refers to padding zeros.

Transformer Net	**2025-01**	



Space–time attention (STA) operation

- ✓ Orange boxes indicate 1 × 1 × 1 convolutions.
- ✓ The soft max function acts on every row of similarity matrix. The shape of corresponding tensors and their proper reshaping are also provided (e.g., $T\times N\times C\rightarrow TN\times C$).
- ✓ $\otimes$ denotes matrix multiplication, and $\oplus$ denotes element-wise sum**.**



Visualized comparison between ViSoT network and others based on transformers.

✓ The four gray circles in the upper right corner serve as references for the size of the parameters

Transformer Net	2025-01	

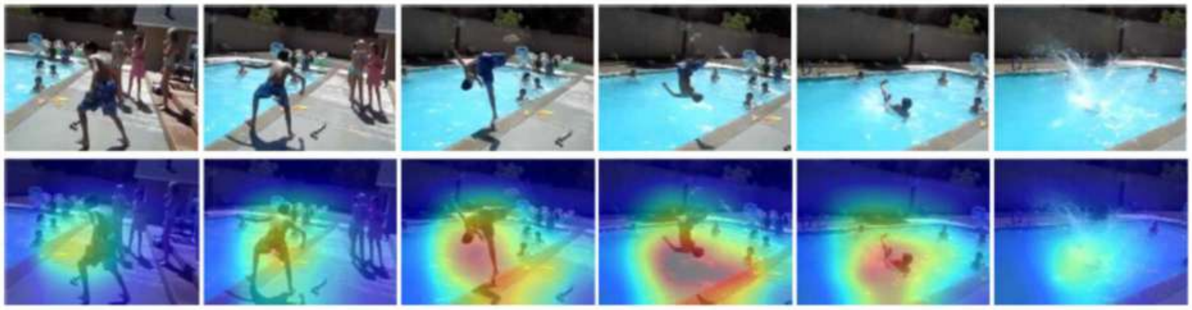


Fig. 8. Visual analysis of ViSoT on the K400 dataset.

Transformer Net	2025-03	Visual explanation

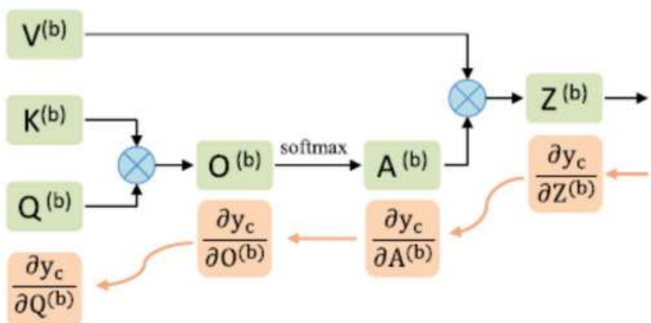


Fig. 2. Diagram of the self-attention module and the gradient back-propagated to the query tensor $\mathbf{Q}^{(b)}$.

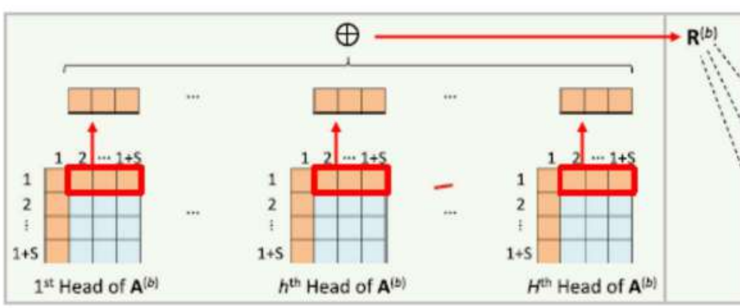


Fig. 3. Diagram of the extraction process for the attention relevance vector $\mathbf{R}^{(b)}$. $\mathbf{R}^{(b)}$ is a general explanations without distinguishing different categories of predictions. The dashed lines in the diagram represent corresponding relationships.

Transformer Net	2025-03	

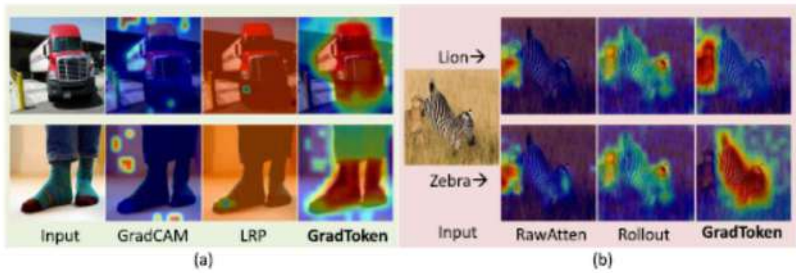


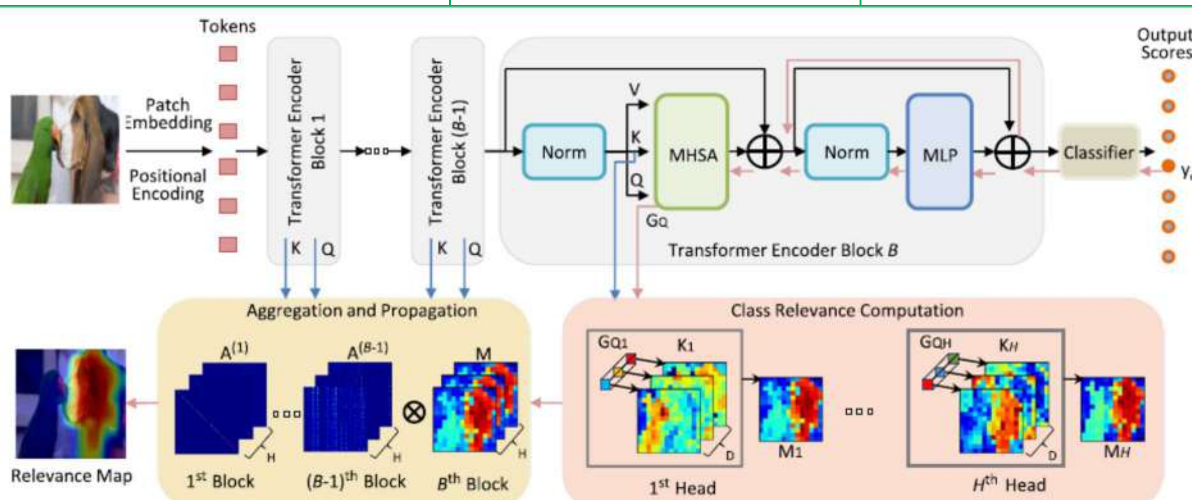
Fig. 1. Visualizations for the model predictions via running the baselines Grad-CAM (Selvaraju, Cogswell, Das, Vedantam, Parikh, & Batra, 2020), LRP (Bach et al., 2015), RawAtten (Clark, Khandelwal, Levy, & Manning, 2019), Rollout (Abnar & Zuidema, 2020), and the proposed GradToken. (a) GradCAM and LRP fail to highlight the target regions. (b) Both RawAtten and Rollout cannot discriminate different targets. In contrast, GradToken can focus on the targets and have better selectivity.

Fig. 1. Visualizations for the model predictions via running the baselines Grad- CAM (Selvaraju, Cogswell, Das, Vedantam, Parikh, & Batra, 2020), LRP (Bach et al., 2015), RawAtten (Clark, Khandelwal, Levy, & Manning, 2019), Rollout (Abnar & Zuidema, 2020), and the proposed GradToken

(a) GradCAM and LRP fail to highlight the target regions.

(b) Both RawAtten and Rollout cannot discriminate different targets. In contrast, GradToken can focus on the targets and have better selectivity

Transformer Net	**2025-03**	



Framework of gradient-decoupling-based token relevance method (GradToken)

🕭 **s-dimensional maps for the convenience of displaying.**

Transformer Net	**2025-03**	

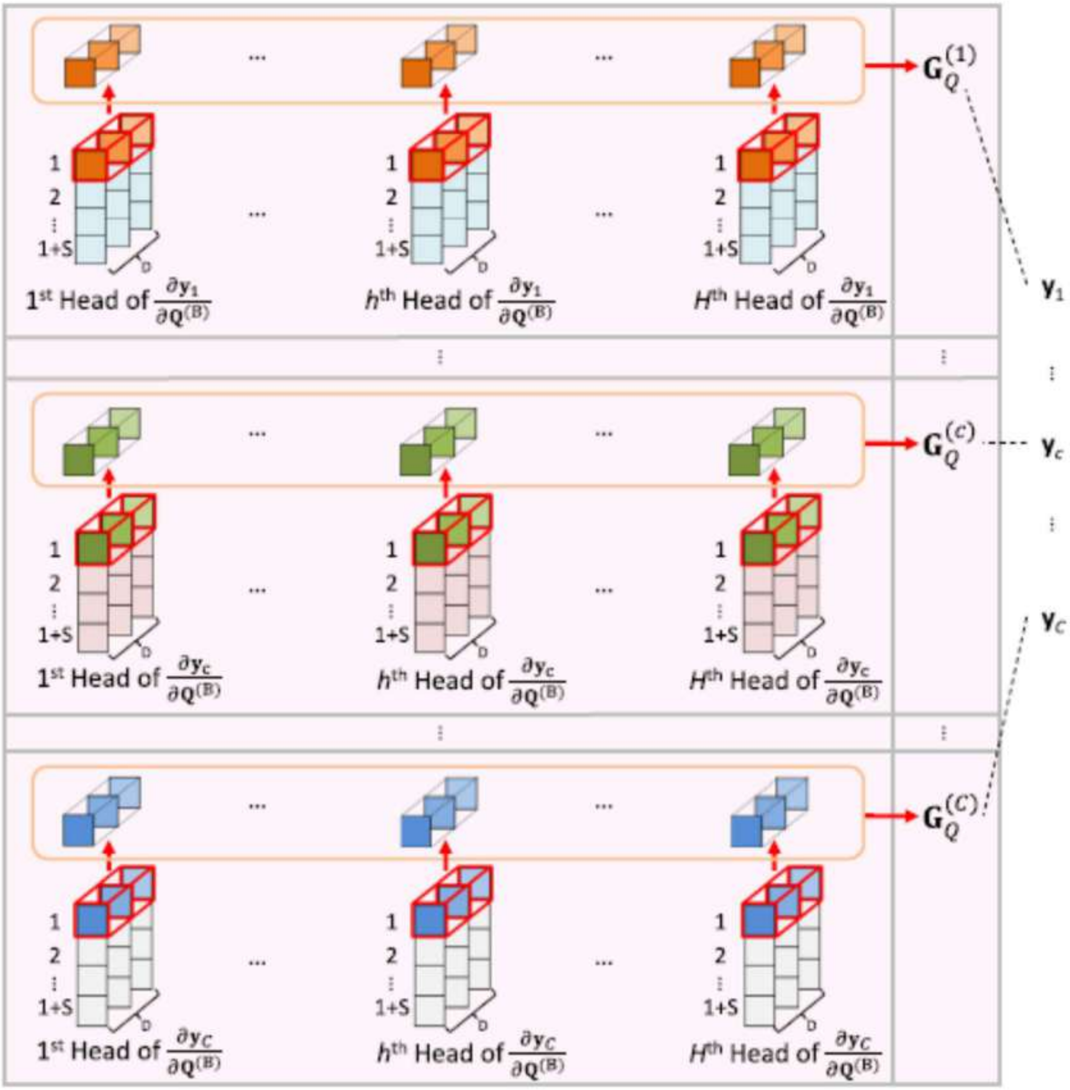
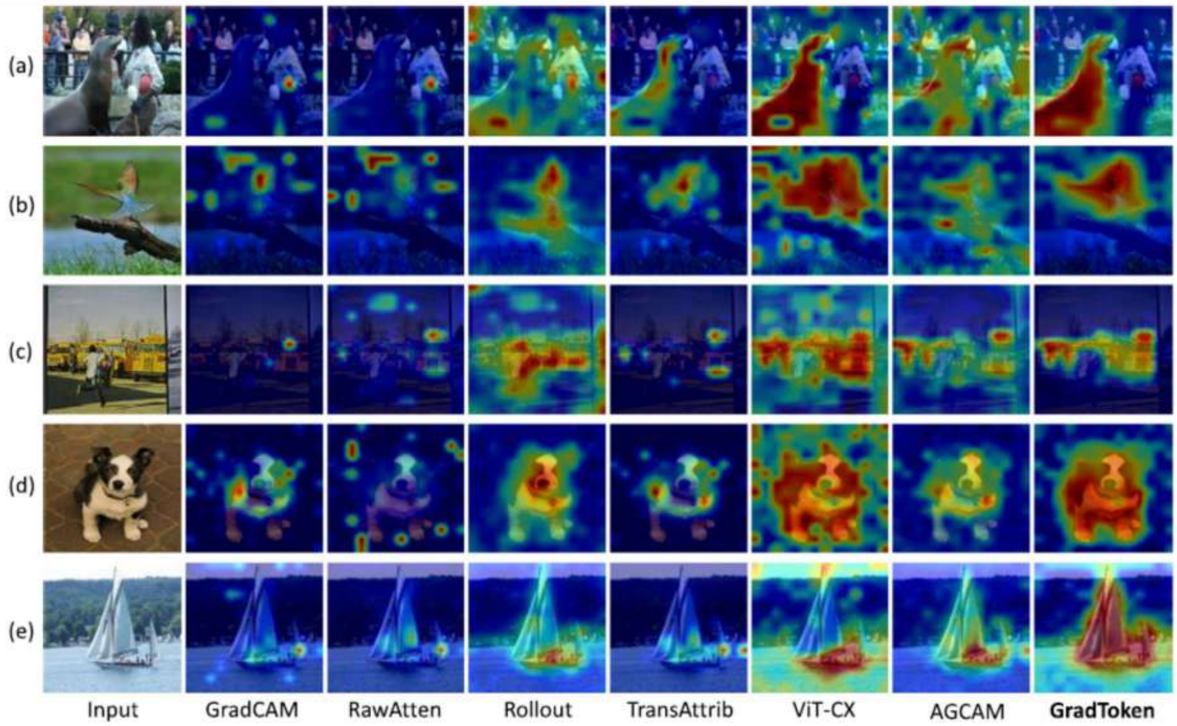


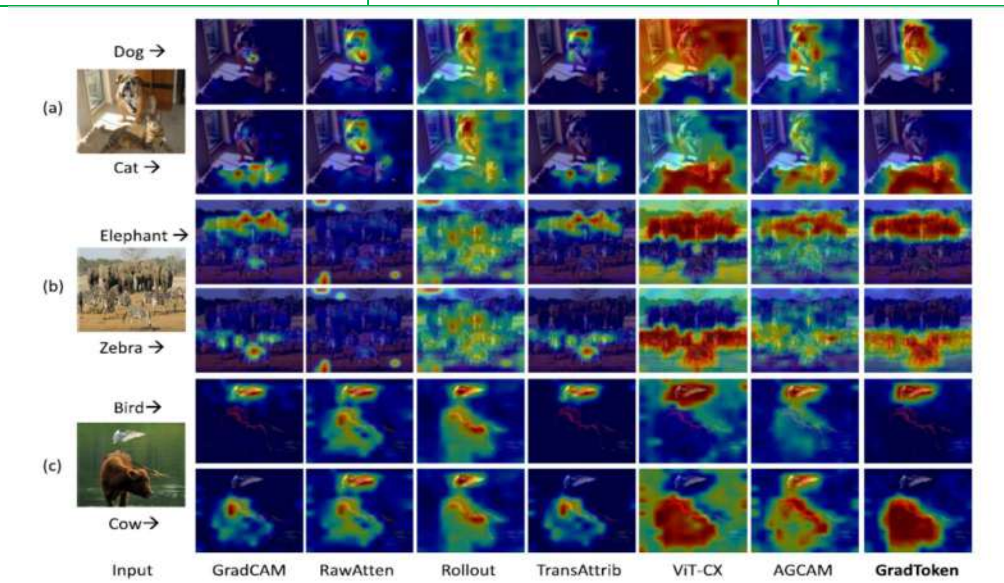
Fig. 5. Diagram of the extraction process for the gradient of the class token $\mathbf{G}_Q$. $\mathbf{G}_Q^{(c)}$ corresponds to the specific target category of prediction $\mathbf{y}_c$. The dashed lines in the diagram represent corresponding relationships.

Transformer Net	2025-03	



Qualitative comparison of the single-class visualizations generated by GradToken and other methods

Transformer Net	2025-03	



Qualitative comparison of the multi-class visualizations generated by the proposed GradToken and other methods.

✓ The texts above and below the input images indicate the target classes of the visualizations

Transformer Net	**2025-03**	

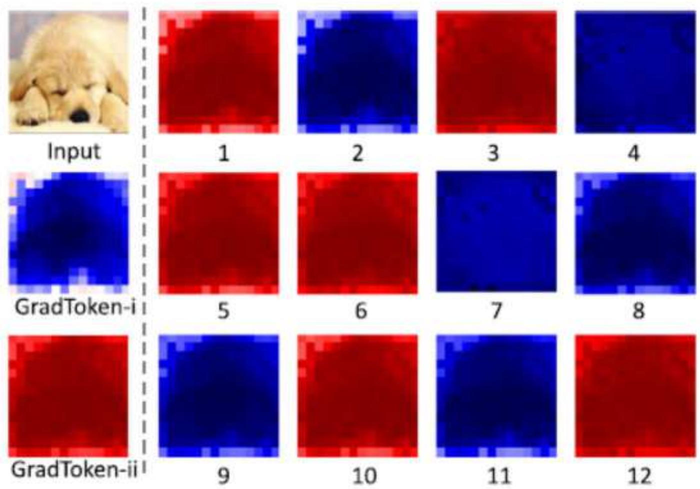
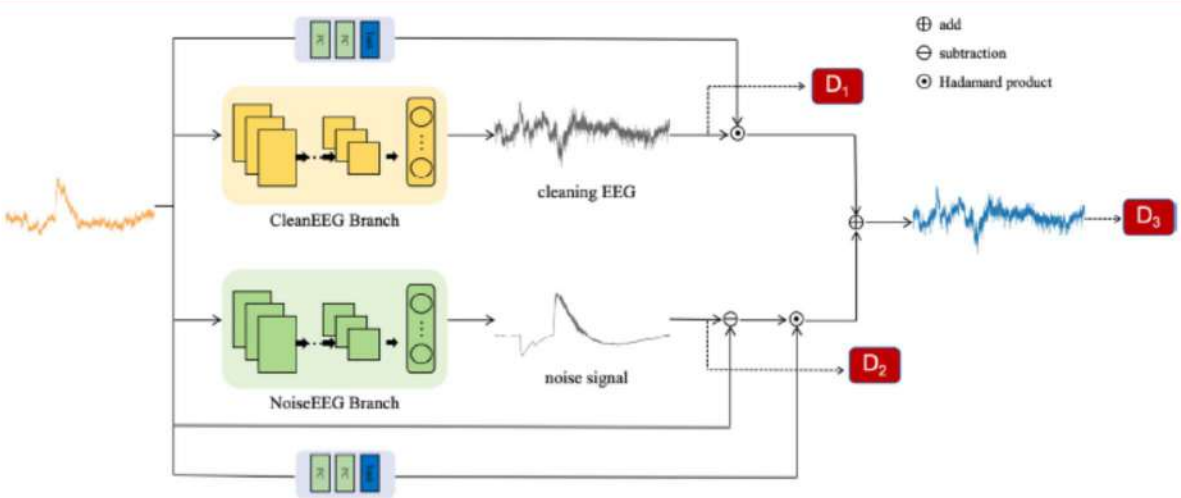


Fig. 10. Multi-head relevance integration and distribution of positive and negative values across different heads. The left side shows the input image and the visualization maps obtained by using two different multi-head integration schemes. The right side shows the visualization maps of twelve heads. The numbers in the figure indicate the head index. Red indicates the positive value and blue indicates the negative value. Some heads with negative values cause the integrated relevance of the target to turn negative in GradToken-i, while having no effect on GradToken-ii.

Transformer Net	**2025-04**	

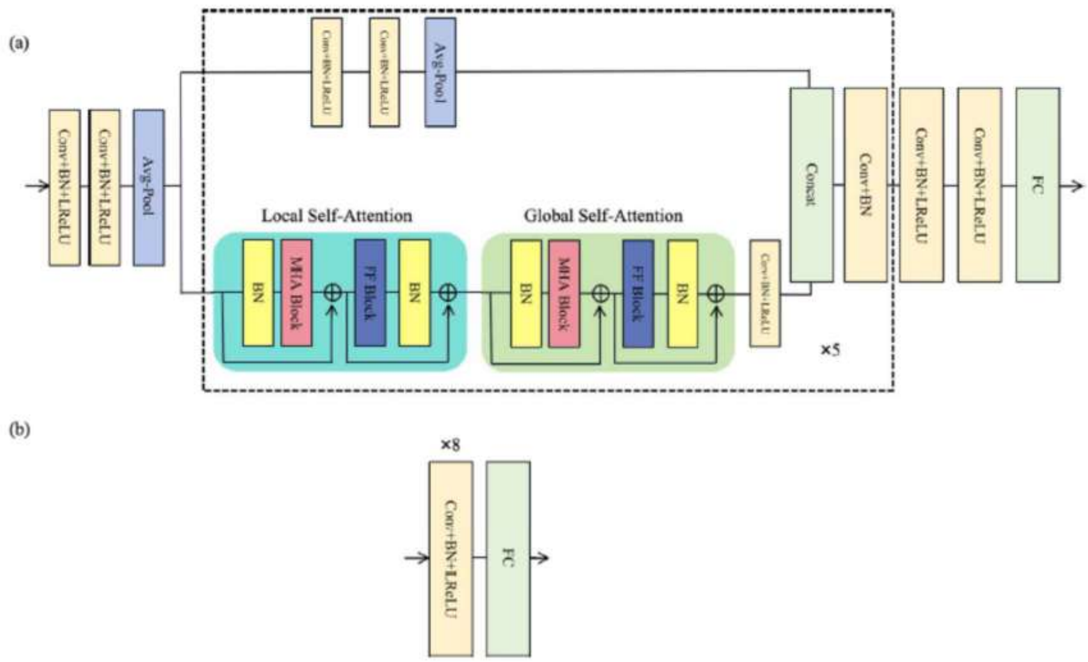


Structure of DHCT-GAN

DHCT: Dual-Branch Hybrid CNN–Transformer Network

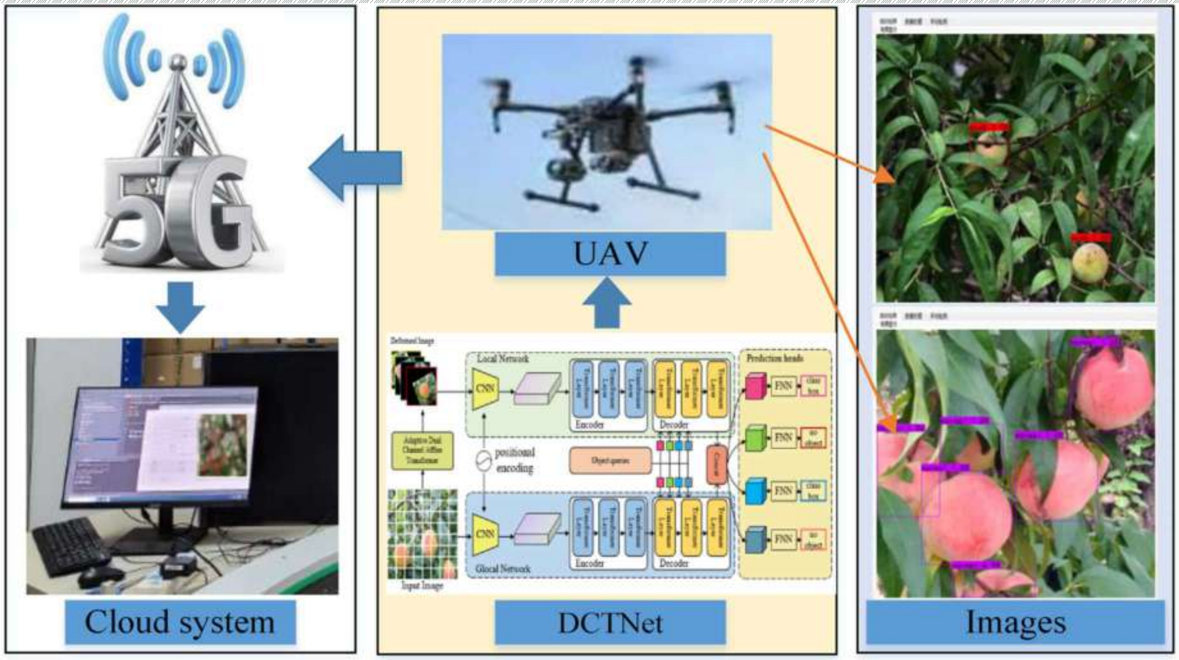
GAN: generative adversarial network

Transformer Net	2025-04	



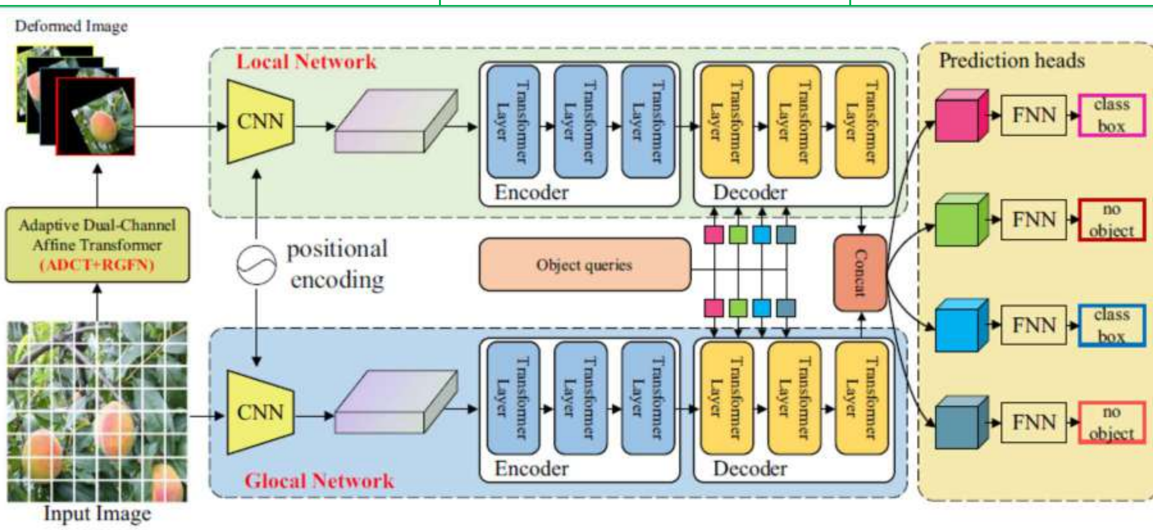
Structure of (a) the generator and (b) the discriminator

Transformer Net	2025-05	



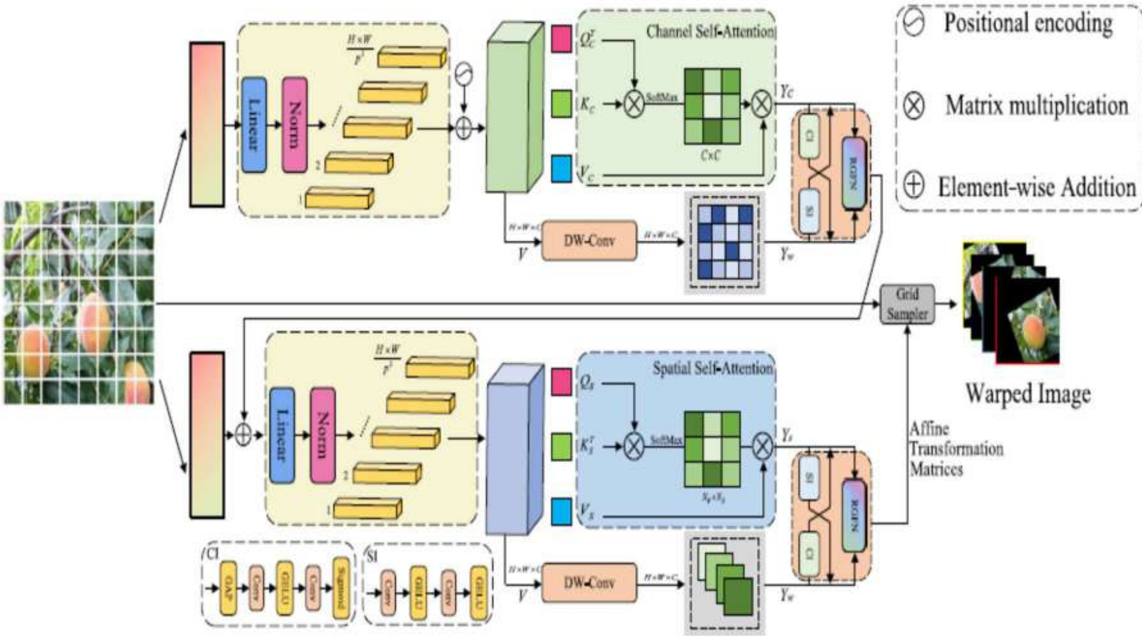
Peach disease detection system

Transformer Net	**2025-05**	

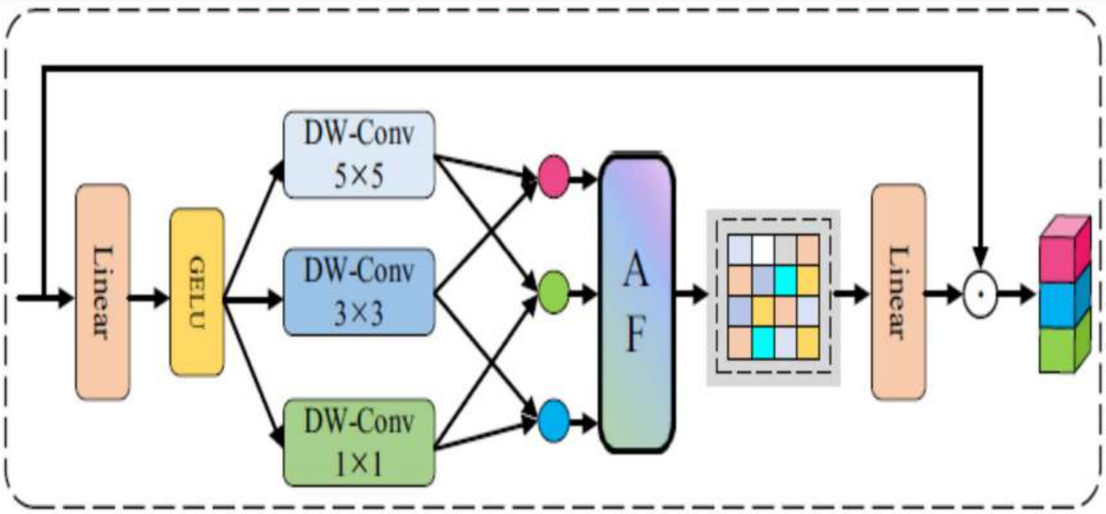


Structure of DCTNet network

Transformer Net	**2025-05**	



Architecture. adaptive dual-channel affine transformer



Architecture. Robust gated feed forward network

Transformer Net	**2025-05**	

Table 3 The hyperparameters in the training process

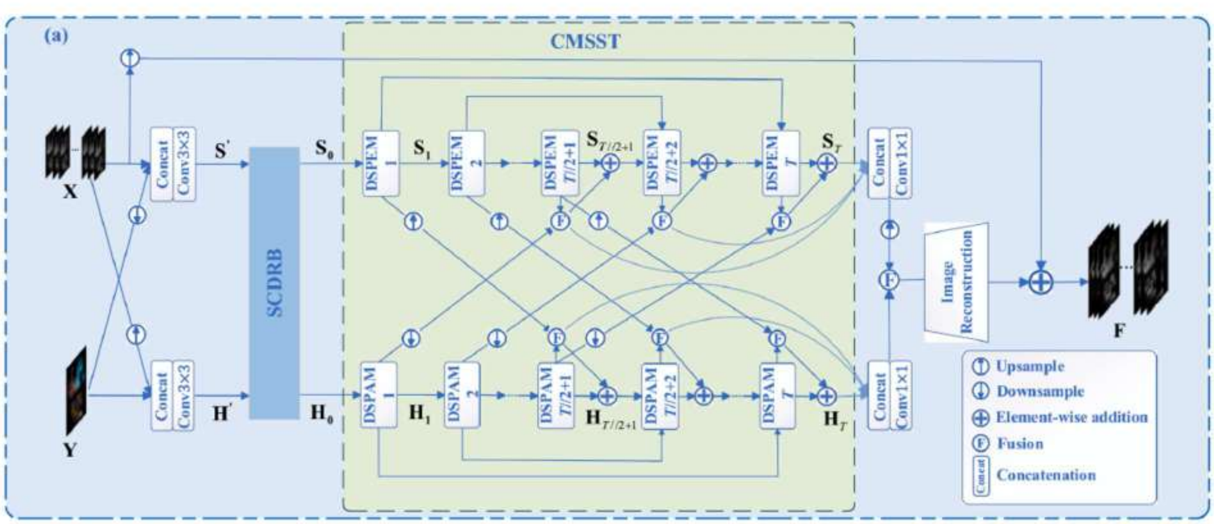
Stage	Hyperparameter	Value
Freeze	Freeze epoch	50
	Freeze batch size	6
UnFreeze	UnFreeze epoch	300
	UnFreeze batch size	4
General	Initial learning rate	1e−3
	Optimizer	SGDM
Image	Size	368KB
	Resolution	512×512

Table 2 Experimental platform setup

Items	Parameters
CPU	Intel(R) Core(TM) i9-12900K
GPU	NVIDIA GeForce RTX 4080
Operating system	Windows 11
Acceleration environment	CUDA 12. 1
Development Platform	Pytorch 1. 7. 0

Transformer Net	**2025-06**	

$\mathbf{X} \in$ $\mathbf{Y} \in$



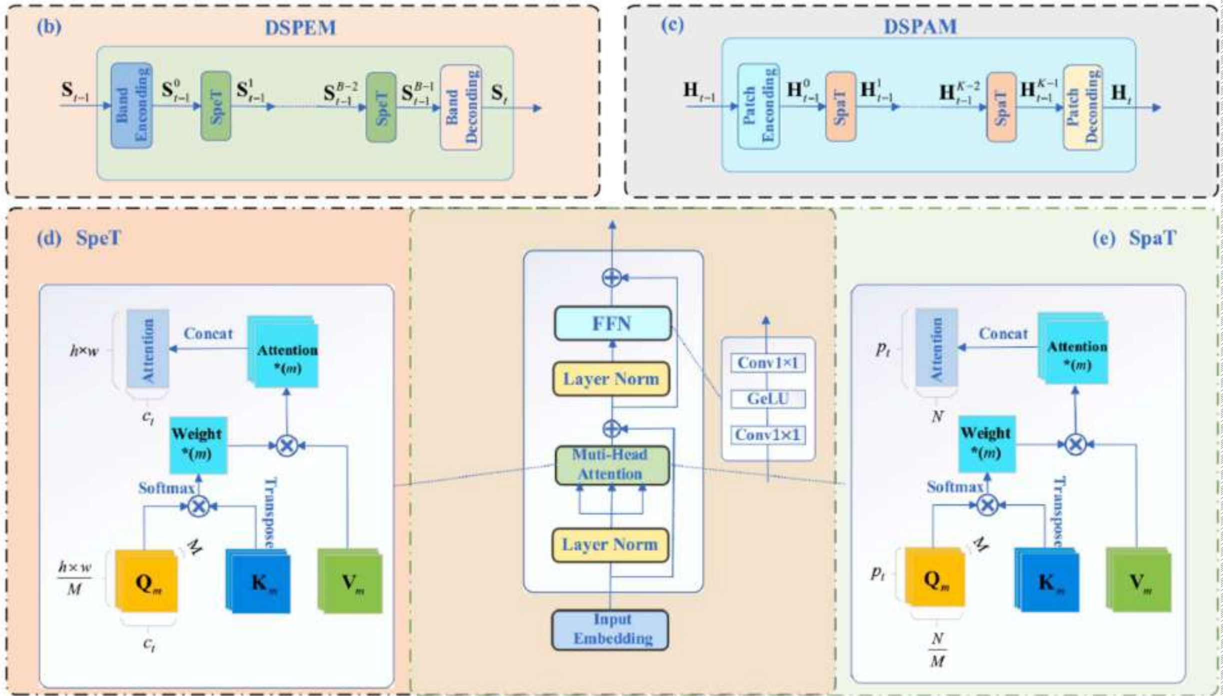


Fig. 1. (a) Architecture of the proposed HCMSST;

- (b) deep spectral-feature-extraction module (DSPEM);
- (c) deep spatial-feature-extraction module (DSPAM);
- (d) structure of the proposed spectral Transformer (spet);
- (e) structure of the proposed spatial Transformer (SpaT).

⊗denotes pixel-wise multiplication operation.

Y. Chen et al. ExpertSystemsWithApplications263(2025)125742

2025-06

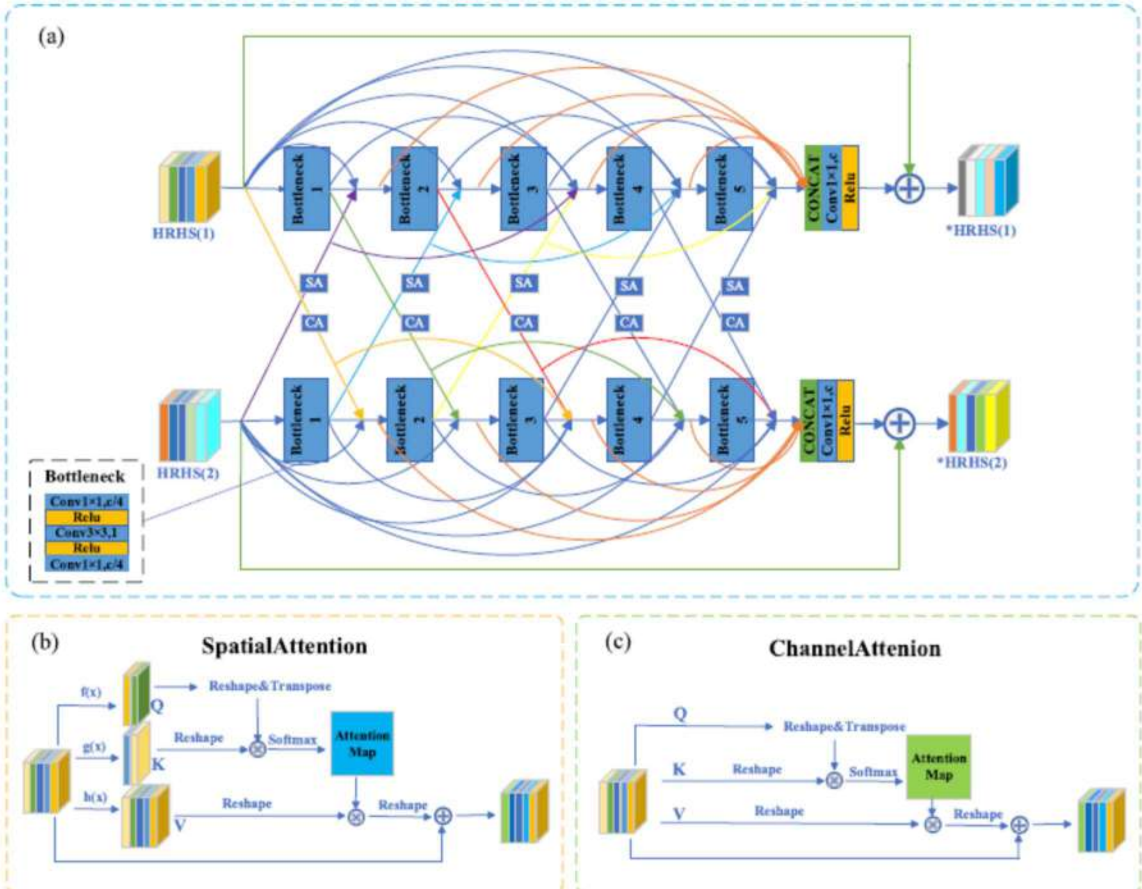


Fig. 2. (a) Structure of the proposed SCDRB; (b) structure of the SA module; (c) structure of the CA module. ⊕ and ⊗ denote the pixel-wise addition and pixel-wise multiplication operations, respectively.

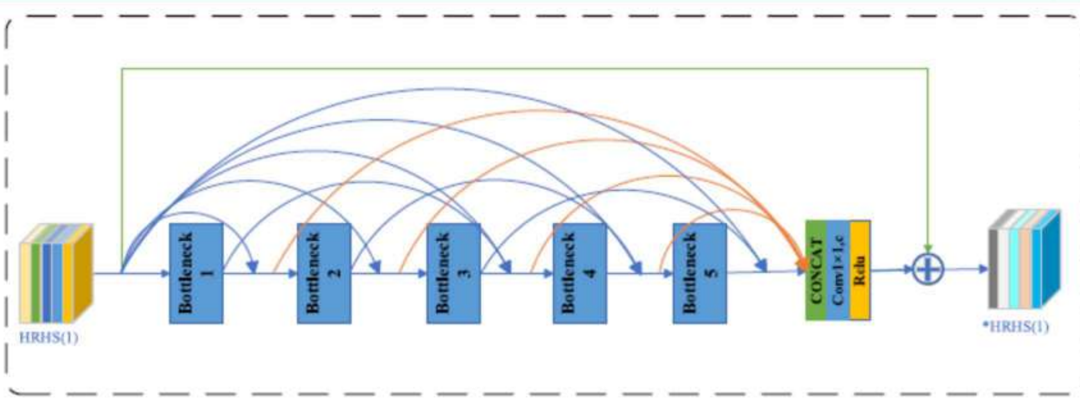
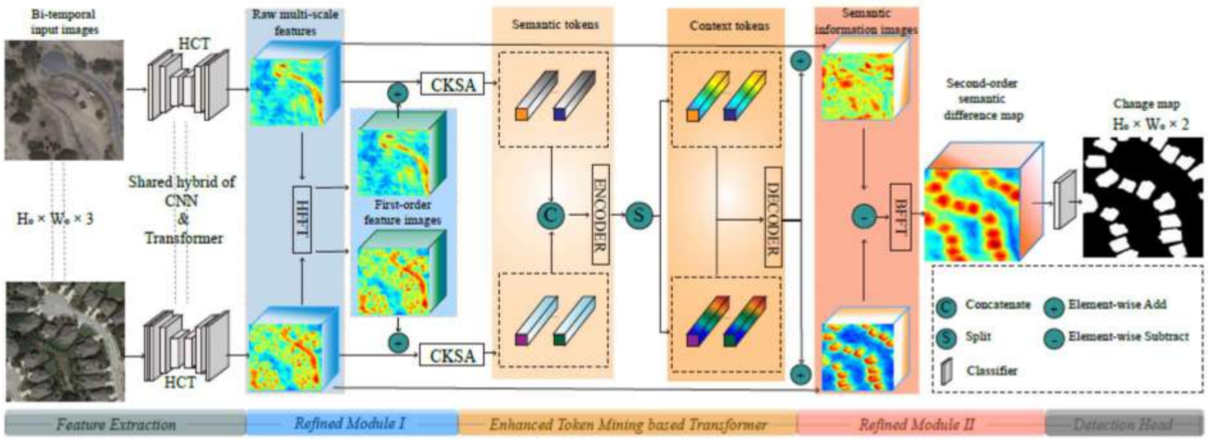


Fig. 3. Structure of the SCRB.

Transformer Net	**2025-07**	



Overall network structure of
Enhanced Hybrid of CNN and Transformer Network (EHCTNet)

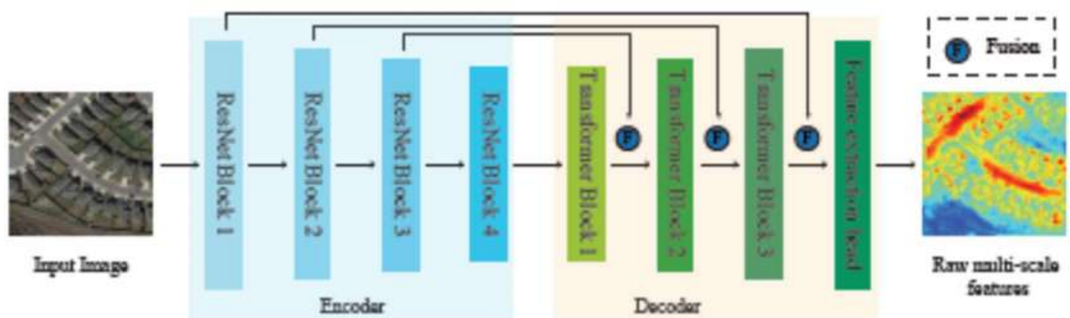


Illustration of the HCT branch

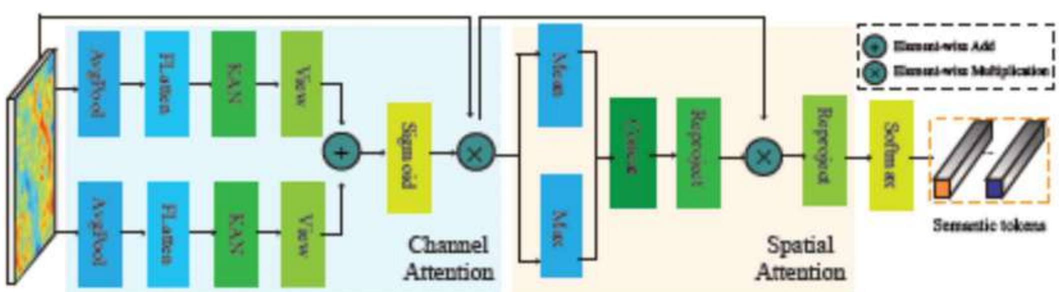
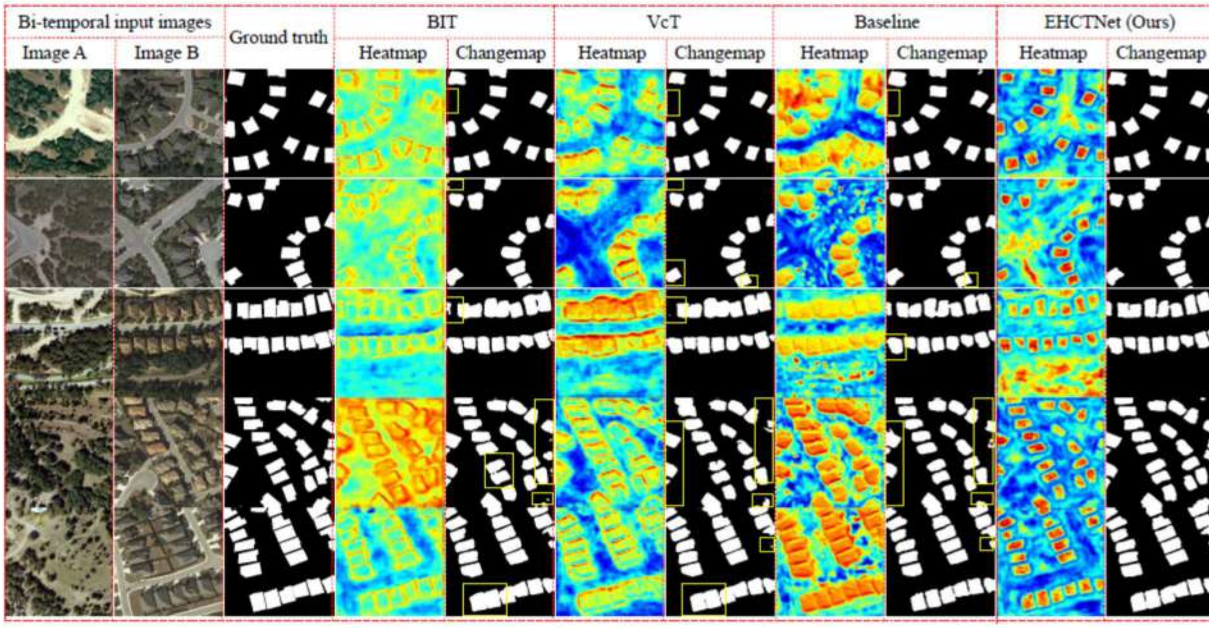


Illustration of CKSA

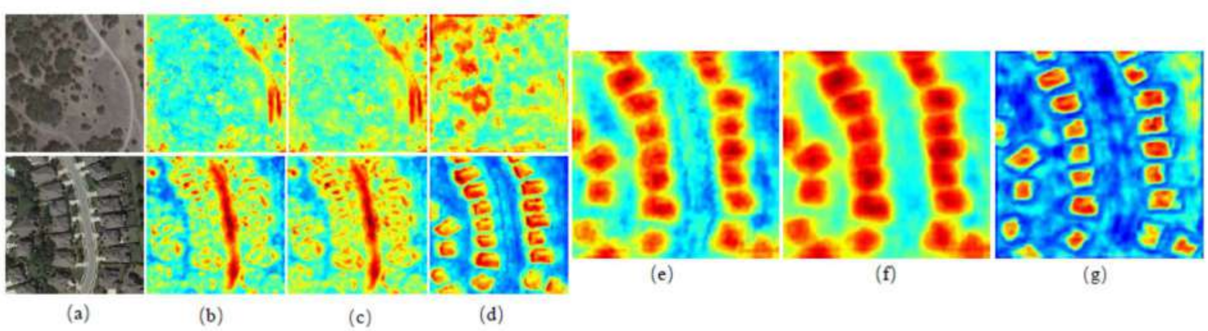
Transformer Net	2025-07	



Heatmaps and change results of our EHCTNet compared with BIT4 & VcT7

- ✓ Baseline: feature extraction module of EHCTNet
- ✓ Yellow rectangle: presence of missed detections, false detections, and incorrect merging of adjacent change areas

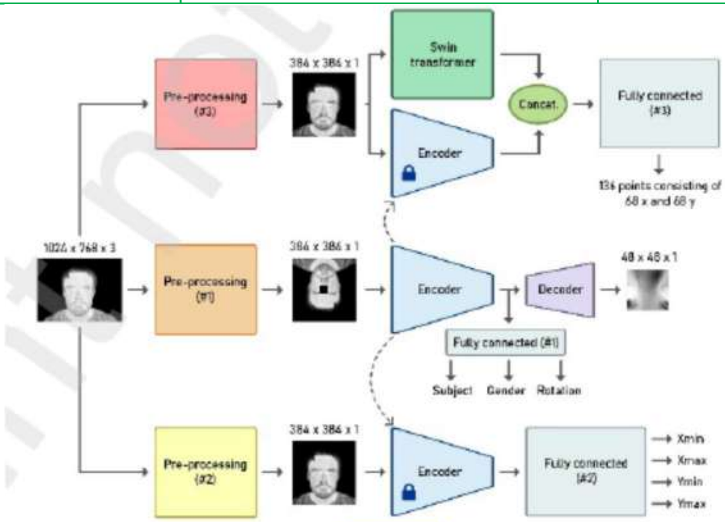
Transformer Net	2025	



Example of network visualization

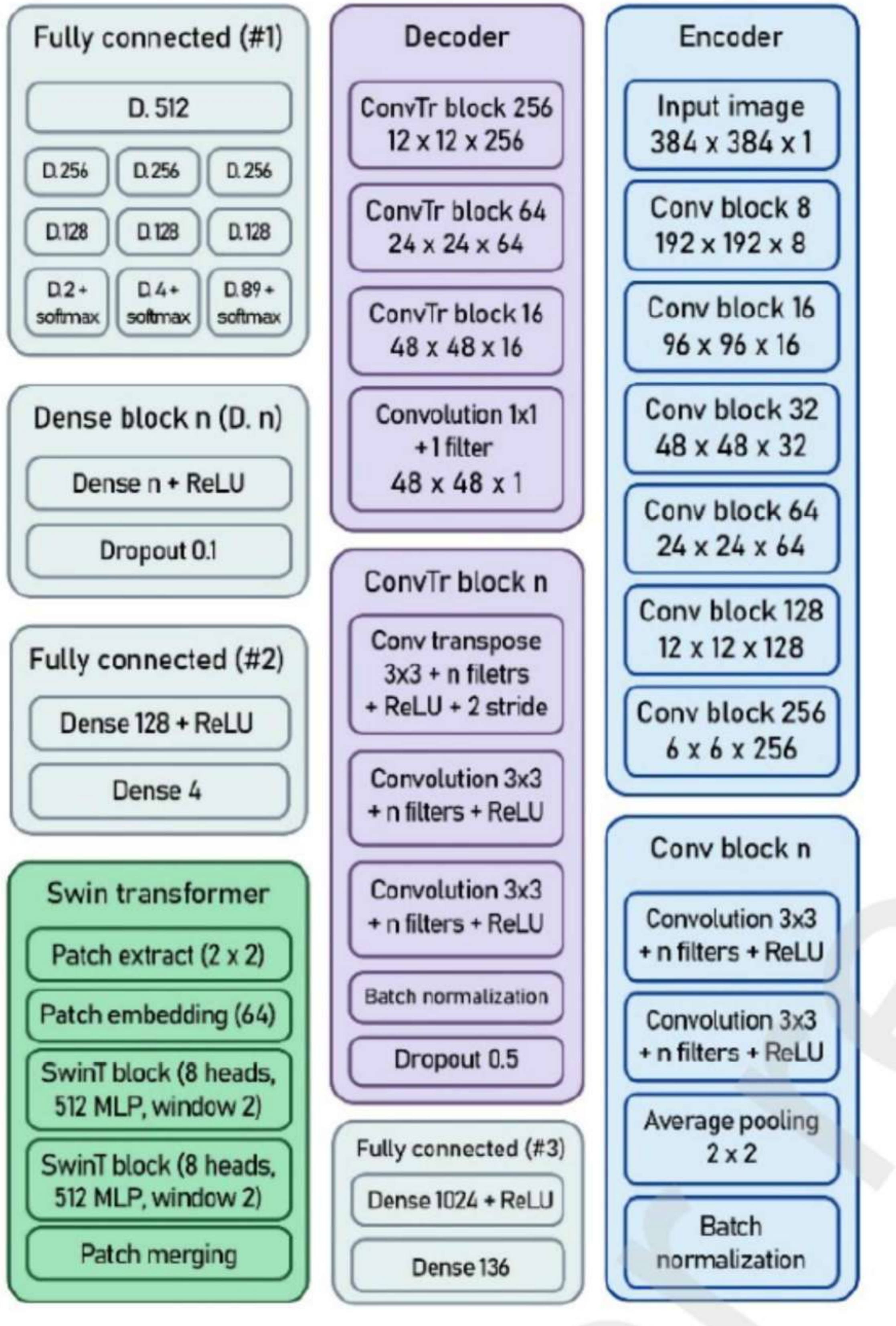
(a): Bi-temporal input images; (b): Raw multi-scale feature images; (c): First-order feature images; (d): Semantic pixel maps; (e): Semantic difference map; (f): Second-order semantic difference map; (g): Change detection heatmap

Transformer Net	2025-08	



Pipeline

- ✓ Initial training on the pretext tasks and representation extraction,
- ✓ Finetuning on target tasks using acquired knowledge.
    - + Swin Transformer further enhances key point detection performance


Fully connected (#1)
D. 512
D.256
D.256
D.256
D.128
D.128
D.128
D.2 + softmax
D.4 + softmax
D.89 + softmax
Dense block n (D. n)
Dense n + ReLU
Dropout 0.1
Fully connected (#2)
Dense 128 + ReLU
Dense 4
Swin transformer
Patch extract (2 x 2)
Patch embedding (64)
SwinT block (8 heads, 512 MLP, window 2)
SwinT block (8 heads, 512 MLP, window 2)
Patch merging
Decoder
ConvTr block 256 12 x 12 x 256
ConvTr block 64 24 x 24 x 64
ConvTr block 16 48 x 48 x 16
Convolution 1x1 + 1 filter 48 x 48 x 1
ConvTr block n
Conv transpose 3x3 + n filetrs + ReLU + 2 stride
Convolution 3x3 + n filters + ReLU
Convolution 3x3 + n filters + ReLU
Batch normalization
Dropout 0.5
Fully connected (#3)
Dense 1024 + ReLU
Dense 136
Encoder
Input image 384 x 384 x 1
Conv block 8 192 x 192 x 8
Conv block 16 96 x 96 x 16
Conv block 32 48 x 48 x 32
Conv block 64 24 x 24 x 64
Conv block 128 12 x 12 x 128
Conv block 256 6 x 6 x 256
Conv block n
Convolution 3x3 + n filters + ReLU
Convolution 3x3 + n filters + ReLU
Average pooling 2 x 2
Batch normalization

Network architectures for pretext tasks, face detection, and keypoint detection

- The encoder is connected to fully connected (#1) for classification and parallelly to the decoder for image inpainting and is trained on these pretext tasks.
- The pre-trained encoder is employed for face detection using fully connected (#2), and for facial keypoint detection, encoder is utilized alongside fully connected (#3) and Swin transformer block

Transformer Net	**2025-09**	

$$\hat{H}_a = f_d(S) \tag{4}$$

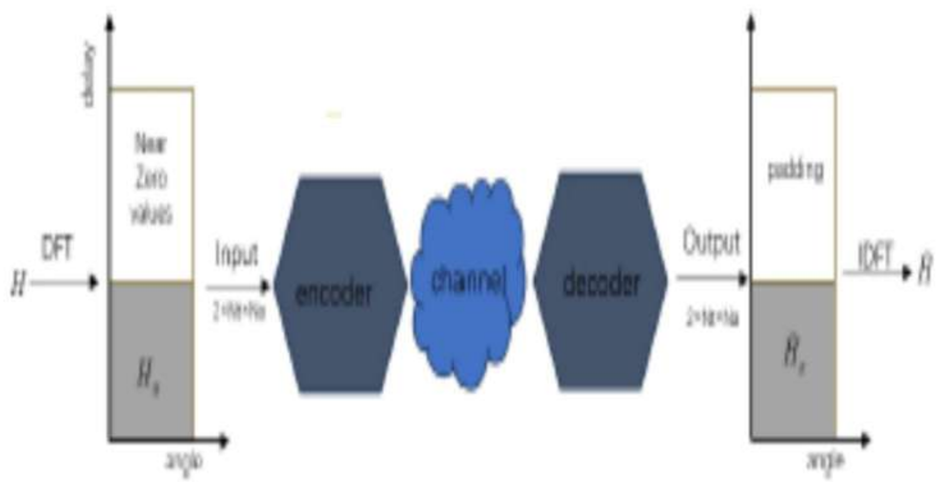
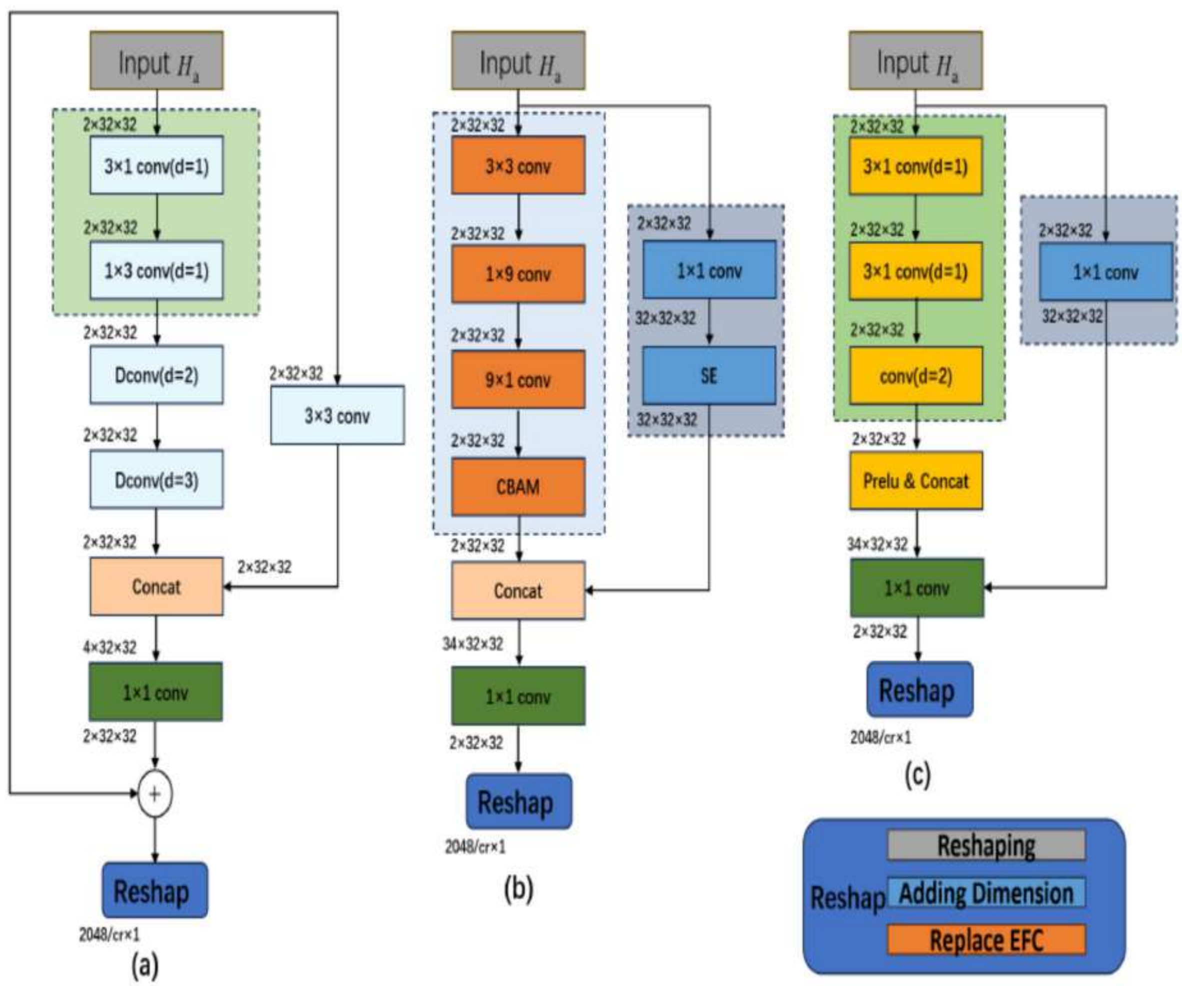
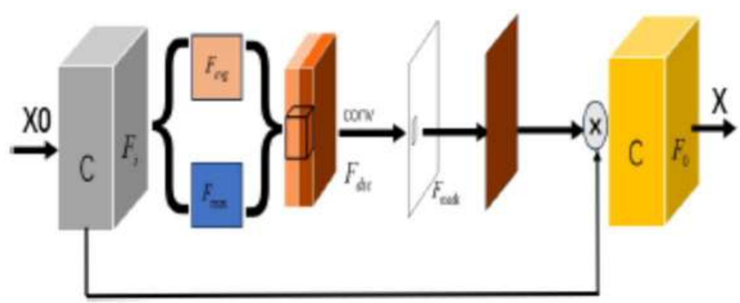


Fig. 1 Overall network architecture.



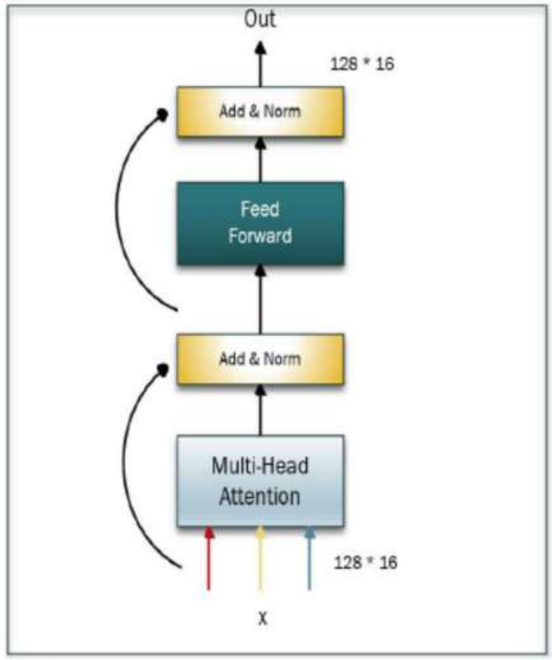
(a),(b),(c), omitting the prelu and LeakyReLU layers for simplicity.

2025-09



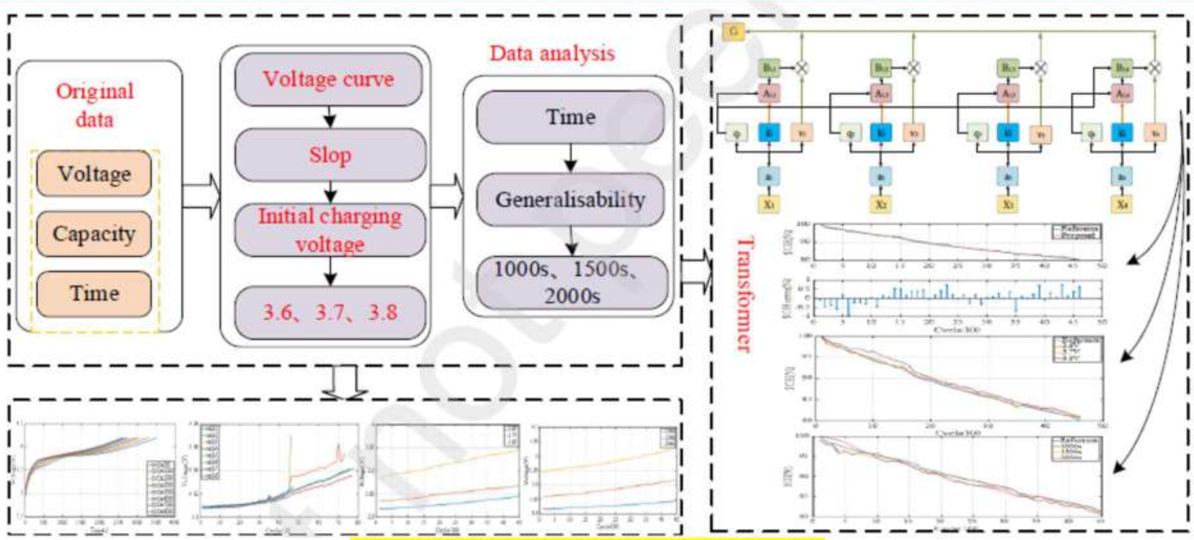
CBAM:The spatial compensation attention mechanism of S-TTNet.

2025-09

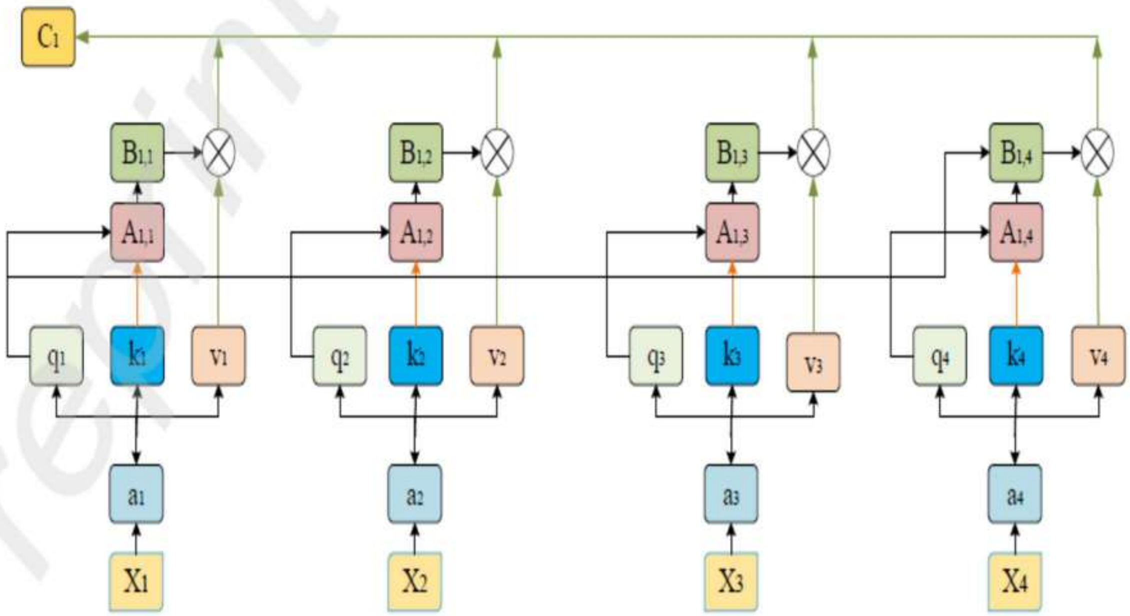


Transformer's self-attention mechanism

Transformer Net	**2025-10**	



STATE OF HEALTH ESTIMATION



Schematic diagram of the multi-head attention mechanism

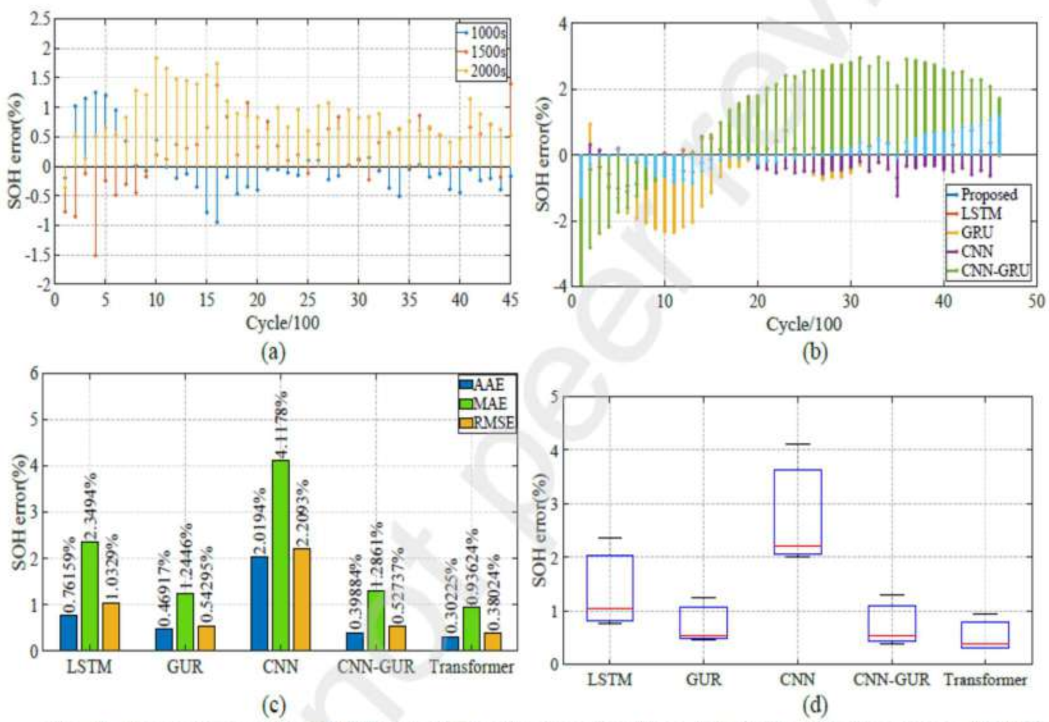
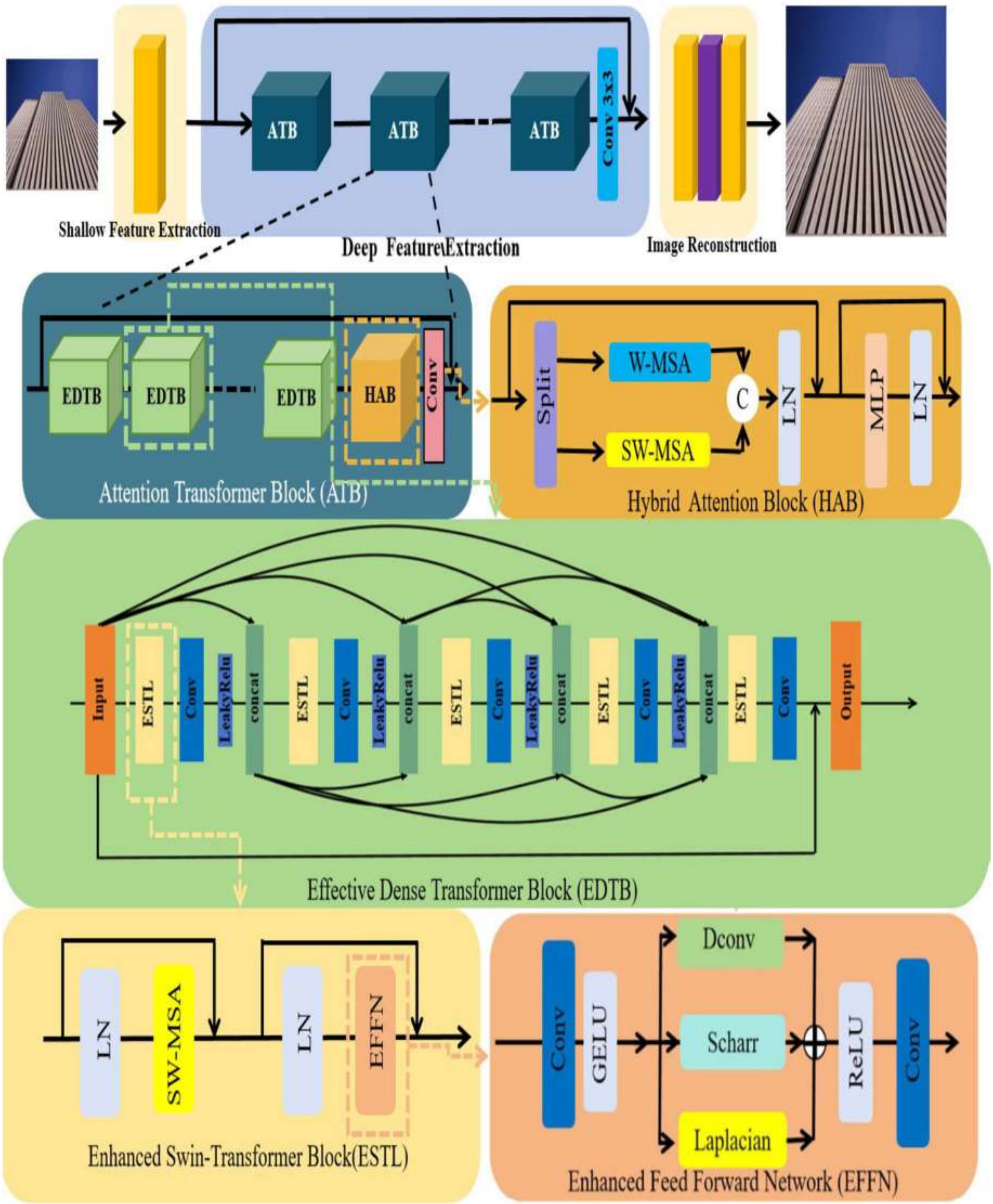


Fig. 7. Comparison results of different SOH estimation algorithms. (a) SOH estimation curves. (b) SOH estimation errors. (c) SOH statistical errors. (d) Error distribution.



Architecture of HADT network

Hybrid Attention-Dense Connected Transformer Networks

Transformer Net	2025-11	

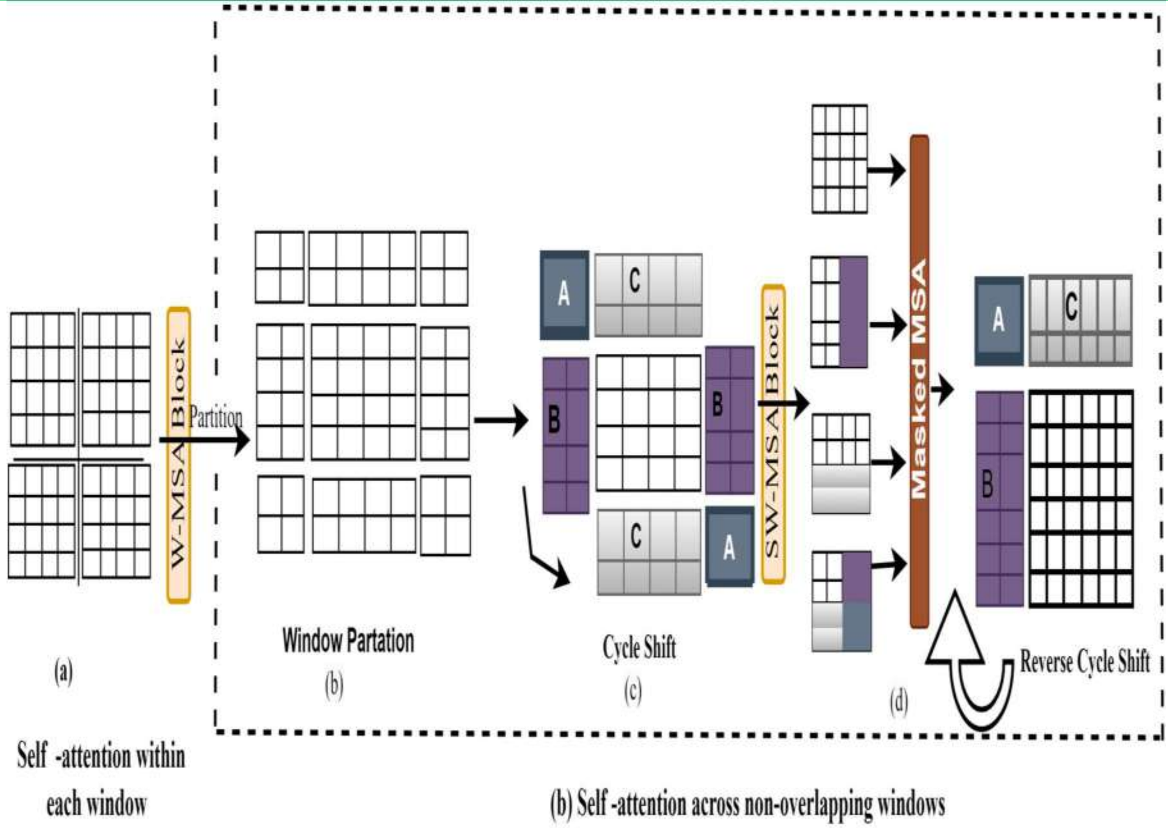
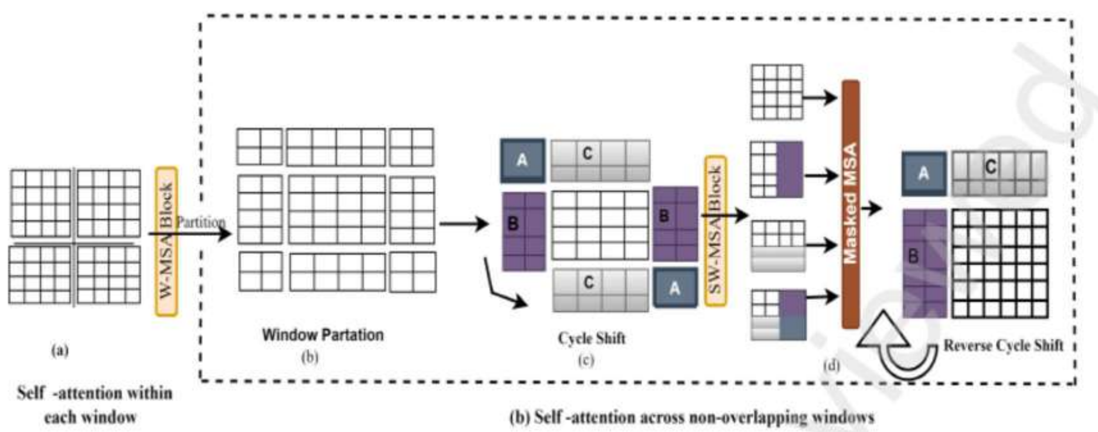
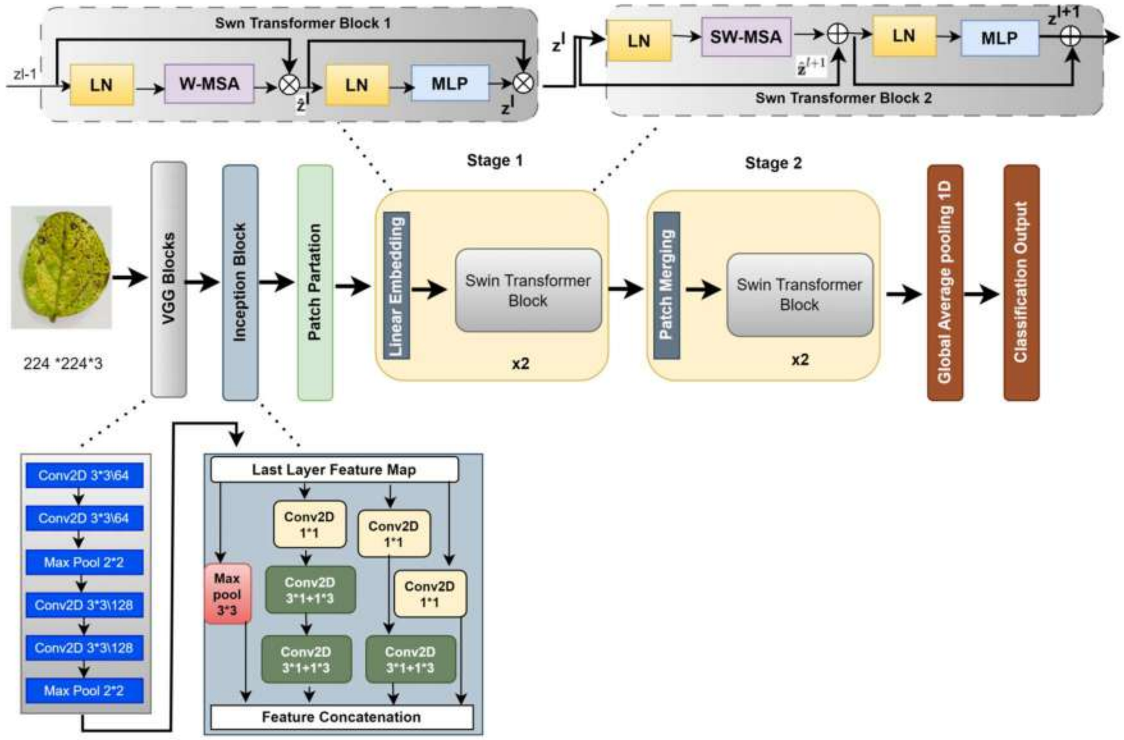


Illustration of W-MSA & SW-MSA

Transformer Net	2025-12	





Architecture

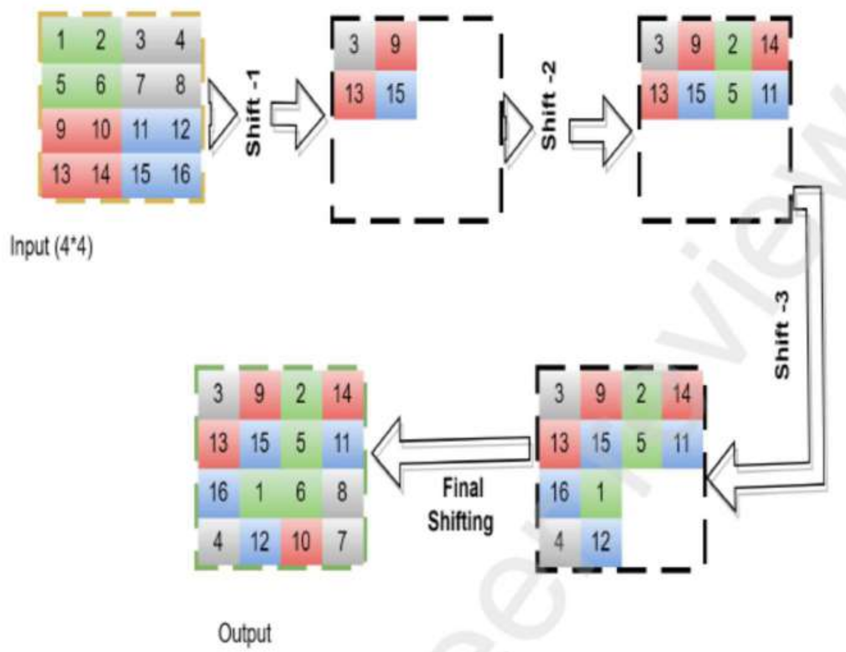


Figure 3: Proposed Shifting operations

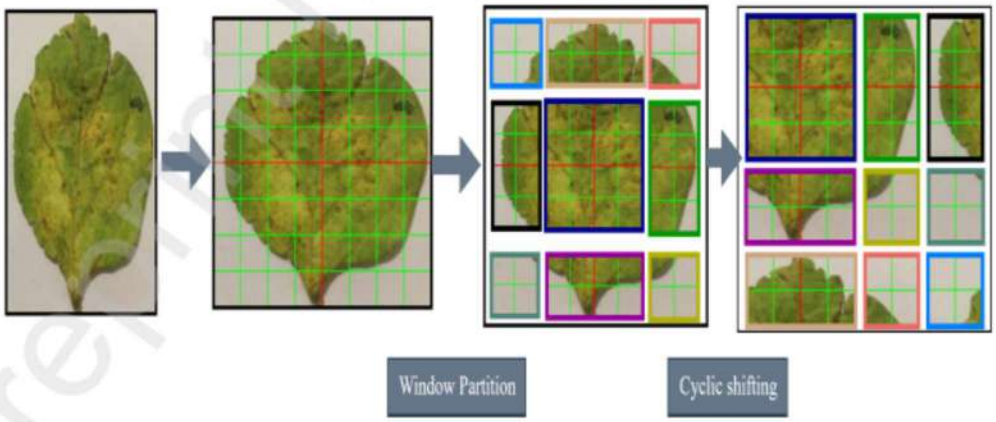


Figure 4: Swin base leaf cyclic shifting

Table 1: Layered description of the proposed model

Type of Layer	Output Vector	Configurations
Input Layer	(None, 224, 224, 3)	Image
VGG Block	(None, 56, 56, 128)	Conv2D (3 x 3) (stride-1), BN, ReLU, MaxPooling2D (2x2) (stride-2)
Inception Block	(None, 56, 56, 512)	Conv2D (diff-diff kernel size and stride), BN, ReLU, MaxPooling2D (2x2) (stride-1)
Stage 1	(None, 14, 14, 512)	Conv2D (4 x 4) (stride-4), LayerNormalisation, W-MSA, MLP, Proposedshifting-MSA
Stage 2	(None, 7, 7, 512)	LayerNormalisation, W-MSA, MLP, Proposedshifting-MSA
Classification	(None, 6)	LayerNormalisation, Avg-Pooling, Linear

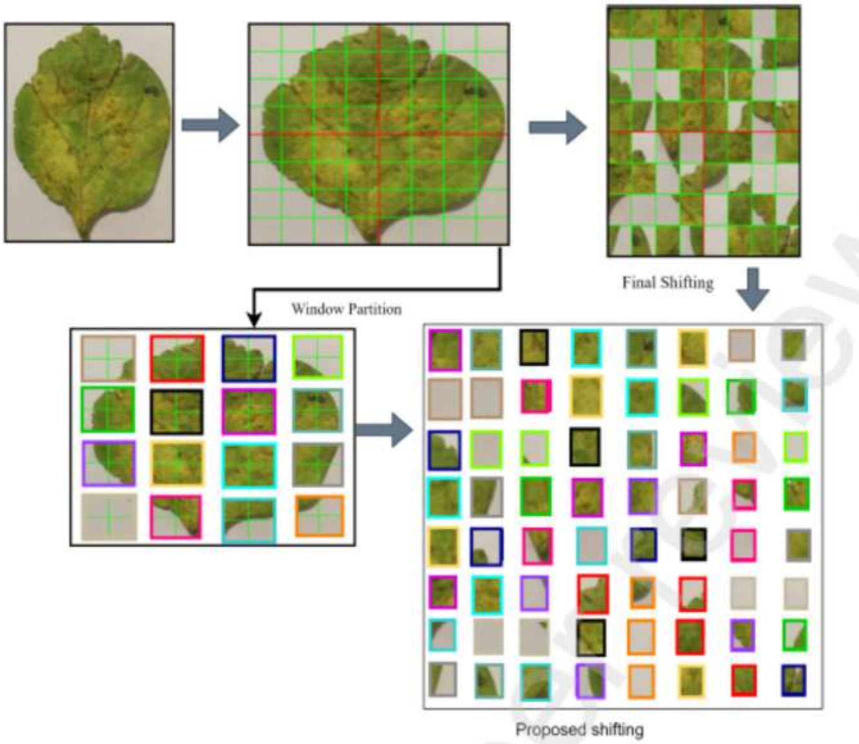


Figure 5: Proposed leaf random shifting

Table 2: Soybean Real Field Dataset

Subject	Information
Type of data	Raw data
How data were acquired	Data curation performed with the help of local farmers, experts, and pathologists using different devices. The Sony Cyber-shot DSC-300 camera, and smartphones, including the Samsung Galaxy M31 equipped with a 64-megapixel camera and an Apple iPhone with a 12-megapixel camera.
Data format	The images are in JPEG format with a standard size of 224×224 pixels.
Description for data collection	The images were captured around 11:30 a.m. and 3:30 p.m. during ideal illumination and mild temperatures for data curation. captured primarily concentrate on the upper regions of soybean leaves affected by insect infestation, as well as leaves that exhibit normal, healthy conditions.
Data source location	Season 1 is collected from Vidisha district, whereas season 2 is from the guna district of Madhya Pradesh, India.

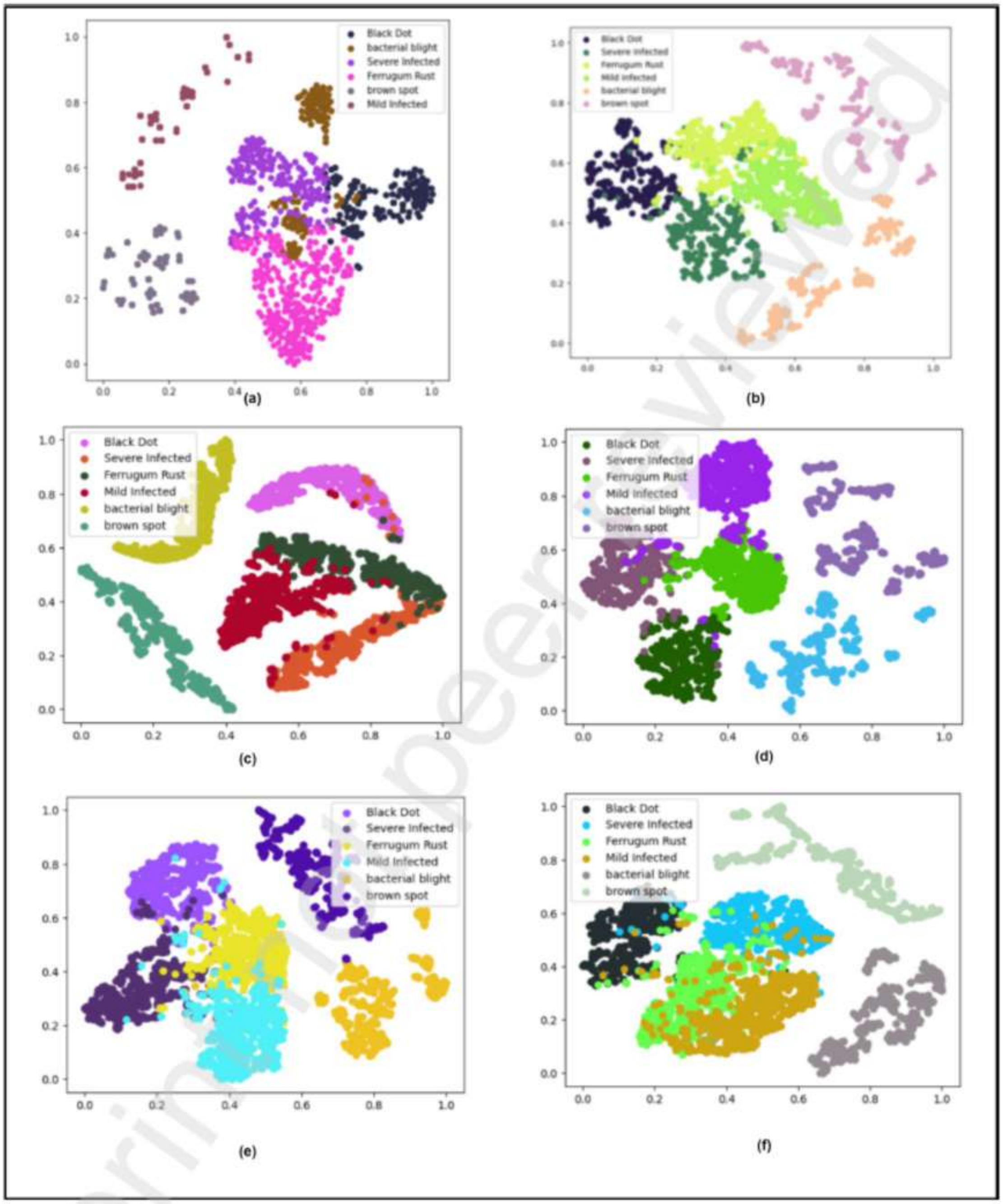


Figure 10: TSNE Plots for soybean dataset (a) Ghost convolution enlightened transformer, (b) PlantXVIT, (c) Convolutional Swin Transformer, (d) Inception convolutional trans formers (ICVT), (e) Former-Leaf, **(f) Proposed SoyaTrans**

Table 4: Accuracy comparison on soybean dataset

Classification Models	Accuracy	Precision	Recall	F-1 score	Parameters in Milions
Ghost-convolution enlightened Transformer [32]	0.67	0.68	0.67	0.66	25.39
PlantXVIT [33]	0.78	0.80	0.78	0.77	6.40
Convolutional Swin Transformer [27]	0.89	0.90	0.88	0.89	27.47
Inception convolutional transformers (ICVT) [34]	0.92	0.96	0.95	0.95	11.16
Former-Leaf [35]	0.74	0.76	0.74	0.74	60.00
Con-Vit [23]	0.73	0.75	0.73	0.75	38.00
RIC-Net [3]	0.88	0.89	0.87	0.88	6.71
MobileNet-V2 [36]	0.80	0.85	0.76	0.80	**5.32**
MFSwin Trans [28]	0.74	0.75	0.74	0.74	12.00
Proposed	**0.94**	**0.93**	**0.94**	**0.92**	**5.20**

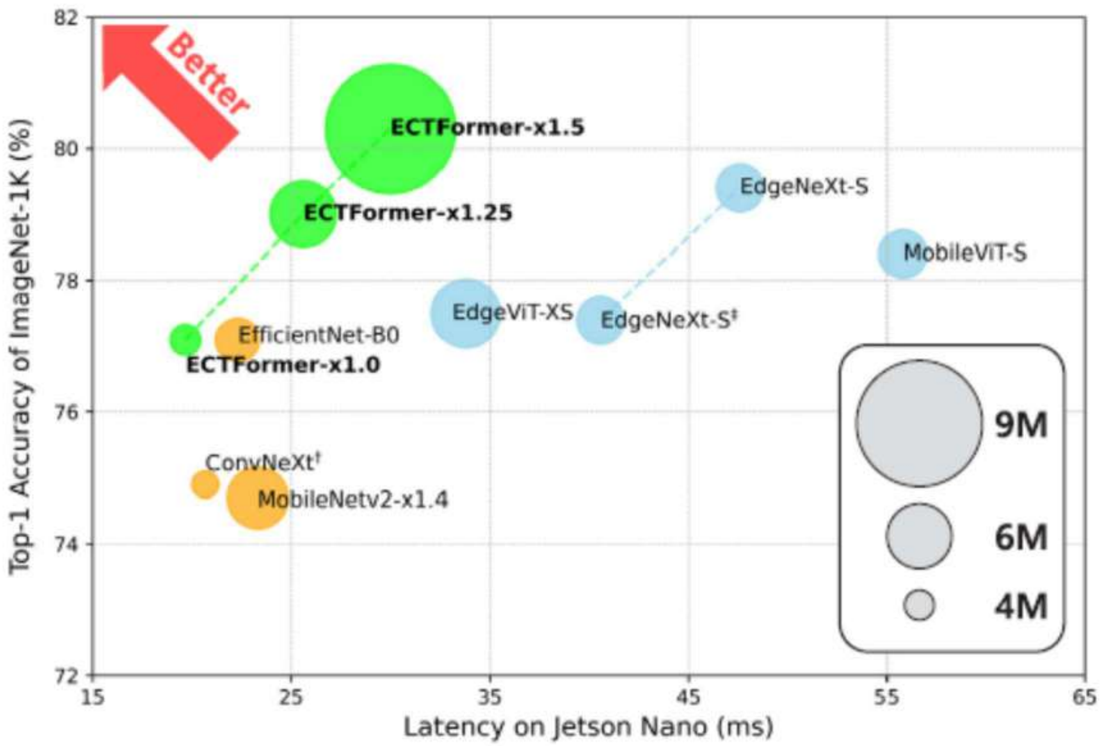
Transformer Net	2025-12	

Algorithm 1 :Proposed Model

```
Input: Dataset
Output: Optimal Classification
Set N_epochs as Number of Epochs
for epoch = 1,2,...,N_epochs do
    Set previous state ← image pixels position
    for batch = 1,2,....T do
        Collect x ∈ P(x) size(224, 224, 3) from dataset to train the model
        current state ← previous state
        window size ← 7x7
        array b have (224,224) position ← 50176
        n_windows ← 1024
        for window =1,2,....n_windows do
            array_c ← elements of current window
            array_b ← array_b -array_c
            array_d ← random 49(7X7) elements from remaining array_b to
            array_b ← array_b - array_d
            array_b ← array_b + array_c
        end for
    end for
end for
Save model after training
```

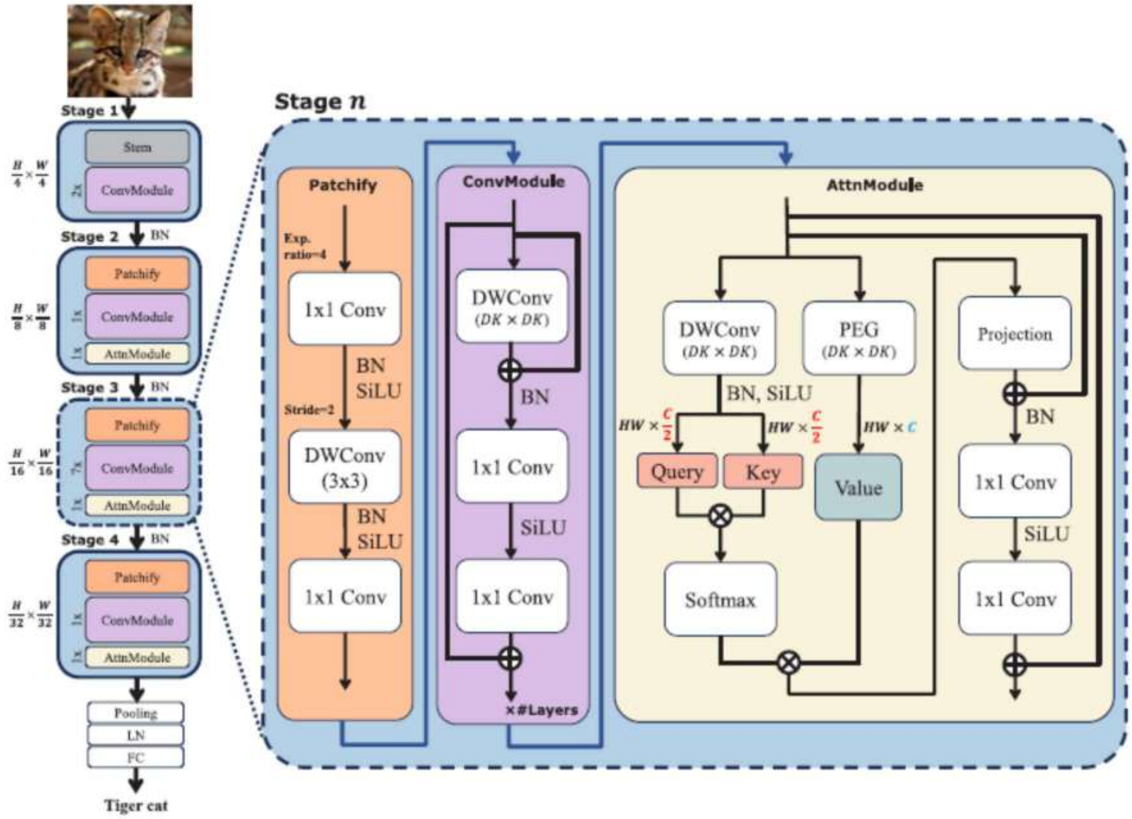
Table 1: Layered description of the proposed model

Type of Layer	Output Vector	Configurations
Input Layer	(None, 224, 224, 3)	Image
VGG Block	(None, 56, 56, 128)	Conv2D (3 x 3) (stride-1), BN, ReLU, MaxPooling2D (2x2) (stride-2)
Inception Block	(None, 56, 56, 512)	Conv2D (diff-diff kernel size and stride), BN, ReLU, MaxPooling2D (2x2) (stride-1)
Stage 1	(None, 14, 14, 512)	Conv2D (4 x 4) (stride-4), LayerNormalisation, W-MSA, MLP, Proposedshifting-MSA
Stage 2	(None, 7, 7, 512)	LayerNormalisation, W-MSA, MLP, Proposedshifting-MSA
Classification	(None, 6)	LayerNormalisation, AvgPooling, Linear

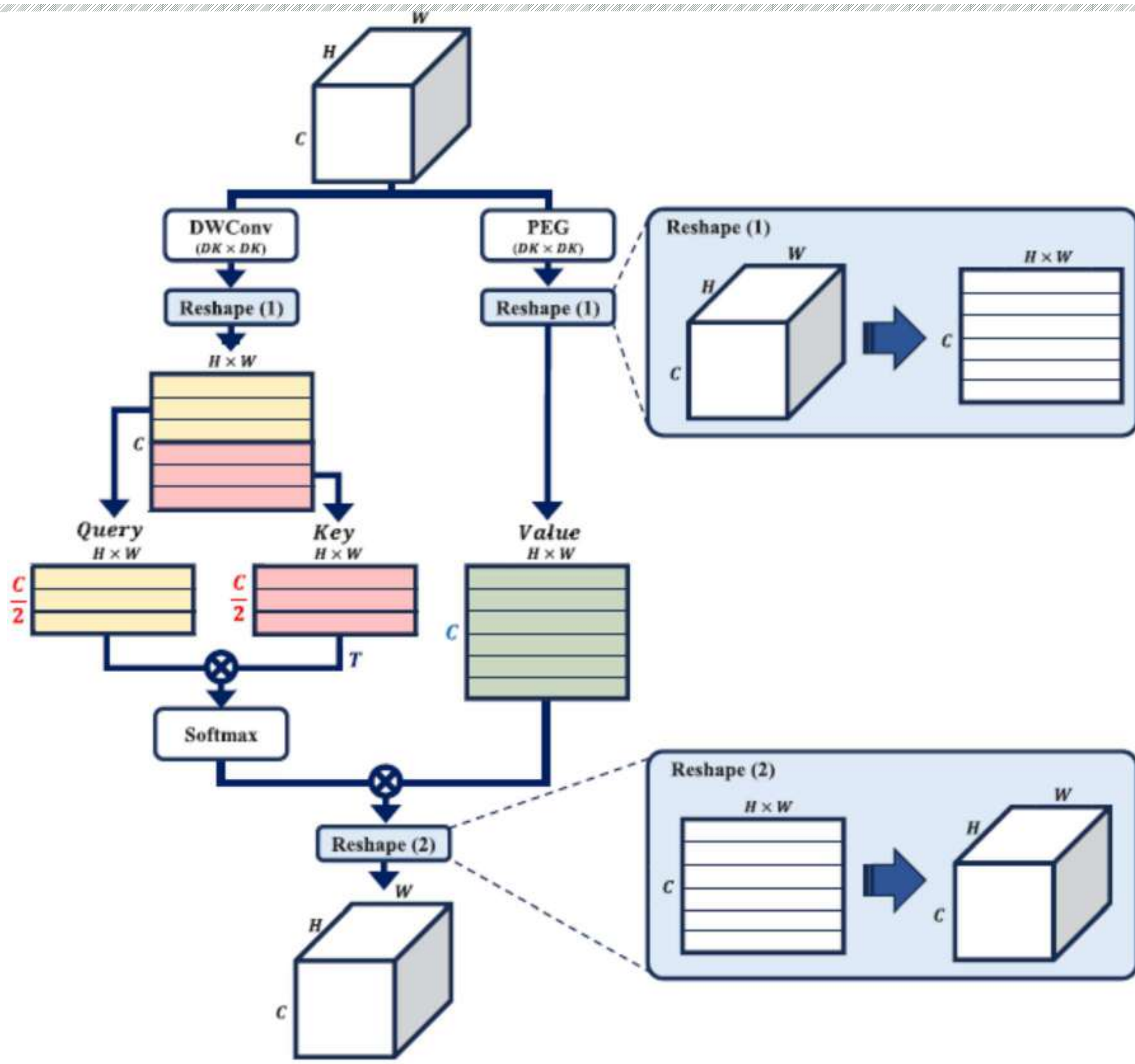


Accuracy vs. Latency on ImageNet-1K

- ✓ Orange and blue color: points indicate performances ConvNets and Conv-Transformer hybrid models
- ✓ Green color: points indicate performances of ECTFormer family
  - + ECTFormer outperforms other lightweight networks with respect to the trade-offs between accuracy and inference speed.



Overall architecture of ECTFormer



Graphical illustration of DSA in AttnModule

- ✓ The input tensor is split to query Q and key K in half along the channel dimensions, which makes prevent an increase in the size of the model parameters.
- ✓ Due to the single-head mechanism, it can effectively alleviate the bottleneck of the attention operation, specifically the reshape operation, resulting in a reduction in the model's execution time

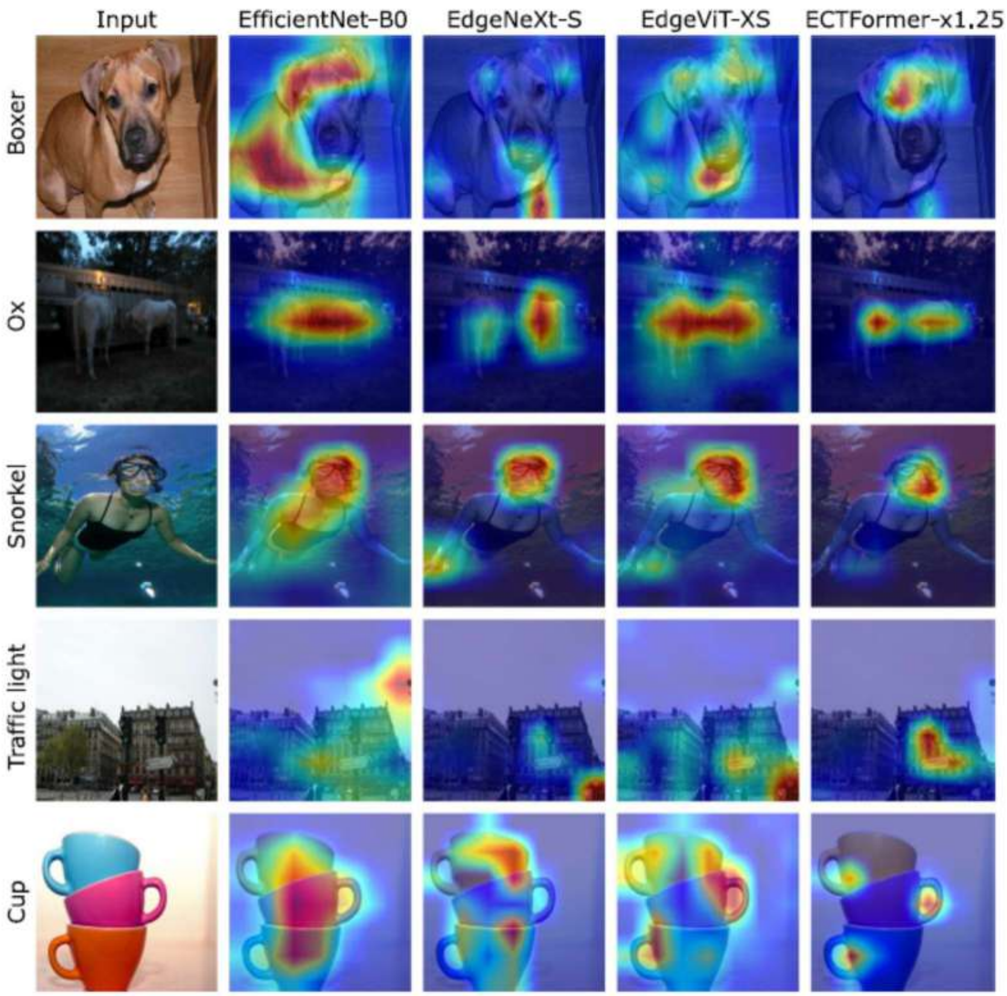
Table 1

Performance comparison of ImageNet-1K.

Models	Params (M)	FLOPs (G)	Size	Top-1 (%)	Throughput	Latency (ms)	Acc./ Latency
MobileNetv2-×1.4 [9]	6.4	0.6	224^2	74.7	4589.1	23.3	3.2
EfficientNet-B0 [32]	5.3	0.4	224^2	77.1	5845.3	22.3	3.5
ConvNeXt[a]	4.1	0.6	224^2	74.9	5586.4	20.7	3.6
MobileViT-S [17]	5.6	1.4	256^2	78.4	2613.2	55.8	1.4
EdgeNeXt-S [13]	5.6	1.3	256^2	79.4	3263.7	47.6	1.7
EdgeNeXt-S[b]	5.6	1.0	224^2	77.3	4207.0	40.6	1.9
EdgeViT-XS [24]	6.8	1.1	224^2	77.5	4551.7	33.8	2.3
ECTFormer-×1.0	4.4	0.7	224^2	77.1	6553.5	19.7	3.9
ECTFormer-×1.25	6.7	1.1	224^2	79.0	4919.8	25.6	3.1
ECTFormer-×1.5	9.6	1.6	224^2	80.3	4244.9	30.0	2.7

[a] Refers to ConvNeXt-Tiny with reduced channel dimensions and depths for each stage, as described in Section 3.4.

[b] Denotes the performance trained with the input image sizes of 224 × 224 without multiscale sampler [17] for a fair comparison.



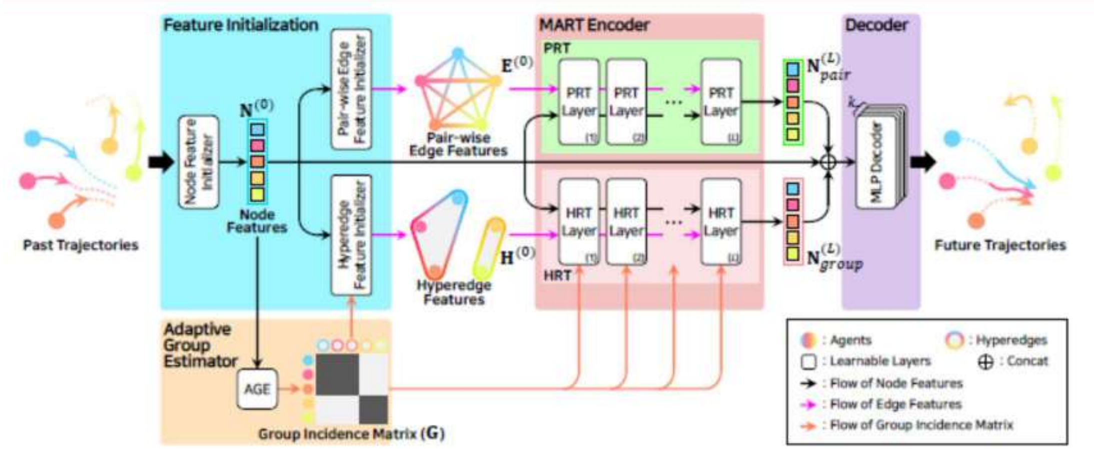
Visualization of the activation maps using Grad-CAM between ECTFormer and the other efficient models

Table 2
Performance comparison on various fine-grained datasets.

Datasets	# Classes	Models	Params (M)	Top-1 (%)	Top-5 (%)	Acc./ Latency
Aircraft [46]	100	ConvNeXt†	3.8	86.5	97.4	4.2
		EfficientNet-B0 [32]	4.1	85.8	96.6	3.8
		EdgeNeXt-S‡	5.3	87.7	96.8	2.2
		EdgeViT-XS [24]	6.4	**88.0**	**97.5**	2.6
		ECTFormer-×1.0	4.1	87.6	97.4	**4.4**
		ECTFormer-×1.25	6.4	87.9	97.1	3.4
Flowers [47]	102	ConvNeXt†	3.8	97.6	99.6	4.7
		EfficientNet-B0 [32]	4.1	98.4	99.7	4.4
		EdgeNeXt-S‡	5.3	97.9	**99.8**	2.4
		EdgeViT-XS [24]	6.4	97.9	99.7	2.9
		ECTFormer-×1.0	4.1	98.3	**99.8**	**5.0**
		ECTFormer-×1.25	6.4	**98.6**	99.7	3.9
Food [48]	101	ConvNeXt†	3.8	87.8	96.9	4.2
		EfficientNet-B0 [32]	4.1	89.0	97.5	4.0
		EdgeNeXt-S‡	5.3	89.4	97.5	2.2
		EdgeViT-XS [24]	6.4	**90.1**	**97.8**	2.7
		ECTFormer-×1.0	4.1	89.5	97.5	**4.5**
		ECTFormer-×1.25	6.4	89.5	97.5	3.5

StanfordCars [49]	196	ConvNeXt†	3.9	92.2	**99.1**	4.5
		EfficientNet-B0 [32]	4.3	91.9	98.8	4.1
		EdgeNeXt-S‡	5.3	92.4	99.0	2.3
		EdgeViT-XS [24]	6.5	**93.1**	**99.1**	2.8
		ECTFormer-×1.0	4.1	92.4	99.0	**4.7**
		ECTFormer-×1.25	6.4	**93.1**	99.0	3.6
Pet [50]	37	ConvNeXt†	3.8	91.6	98.7	4.1
		EfficientNet-B0 [32]	4.1	91.8	98.7	3.8
		EdgeNeXt-S‡	5.3	93.1	99.0	2.1
		EdgeViT-XS [24]	6.4	92.2	99.1	2.5
		ECTFormer-×1.0	4.1	92.5	99.0	**4.4**
		ECTFormer-×1.25	6.3	**93.2**	**99.2**	3.4
Caltech-256 [51]	256	ConvNeXt†	3.9	84.5	93.9	3.6
		EfficientNet-B0 [32]	4.3	85.4	94.4	3.5
		EdgeNeXt-S‡	5.4	86.7	94.8	1.9
		EdgeViT-XS [24]	6.5	84.7	94.1	2.3
		ECTFormer-×1.0	4.2	86.4	94.9	**3.9**
		ECTFormer-×1.25	6.4	**87.5**	**95.1**	3.1

Transformer Net	**2025-15**	



MART: Model architecture of MultiscAle Relational Transformer network

Four components

- ✓ Feature initialization (sky blue), Adaptive Group Estimator (orange), MART encoder (red), and future trajectory decoder (Decoder) (purple)
  - o MART encoder includes a pair-wise relational transformer (green) and a hyper relational transformer (pink).

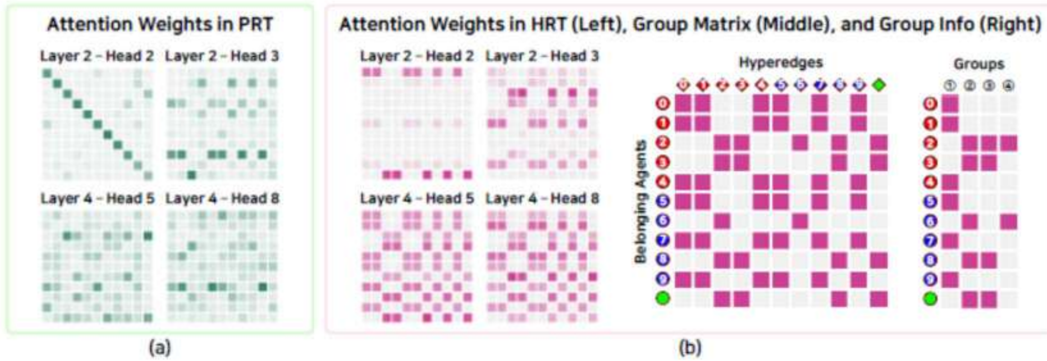


Fig. 7: (a) Visualization of attention weights in PRT. (b) Visualization of attention weights in HRT (left), the group incidence matrix G (middle), and the corresponding groups (right). The sample results from the identical test data sample used in Figure 6. For the attention visualization, min-max normalization is applied to ensure a clearer visual understanding, and darker colors indicate higher attention weights.

Transformer Net	2025-16	

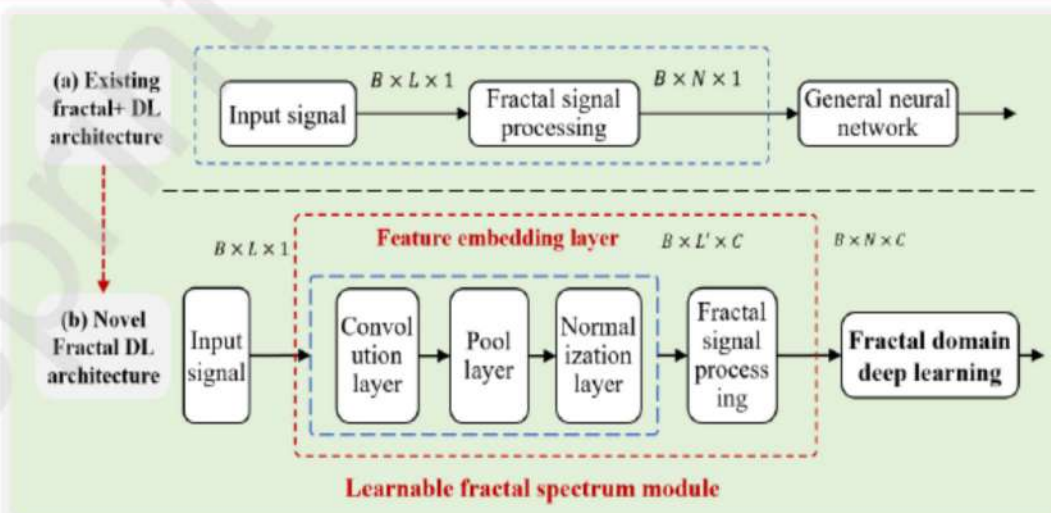


Figure 1. Comparative Diagram: Existing Models *vs.* fractal domain deep learning architecture

Transformer Net	2025-16	

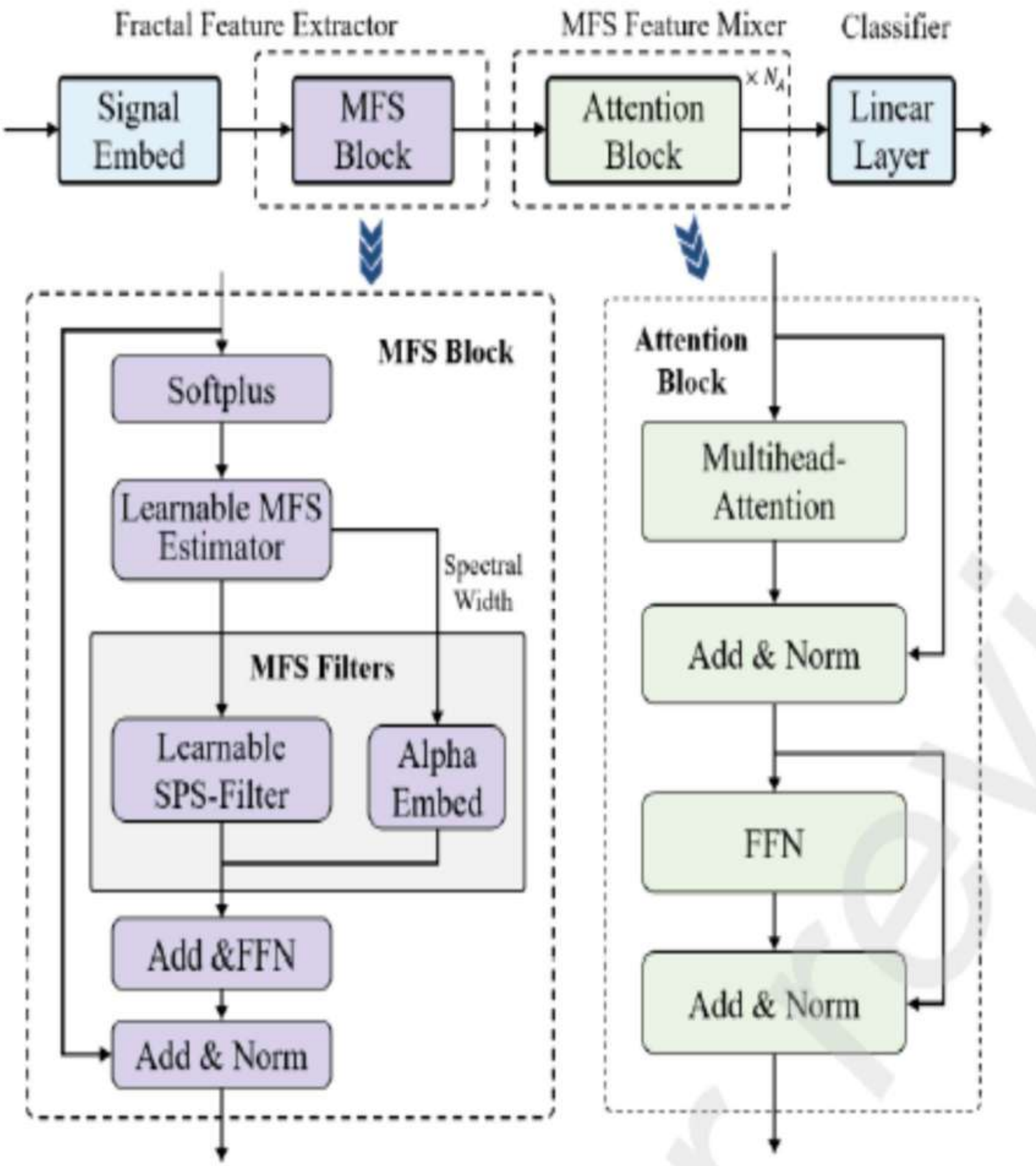


Figure 2. Overall structure of MFS-Transformer Network

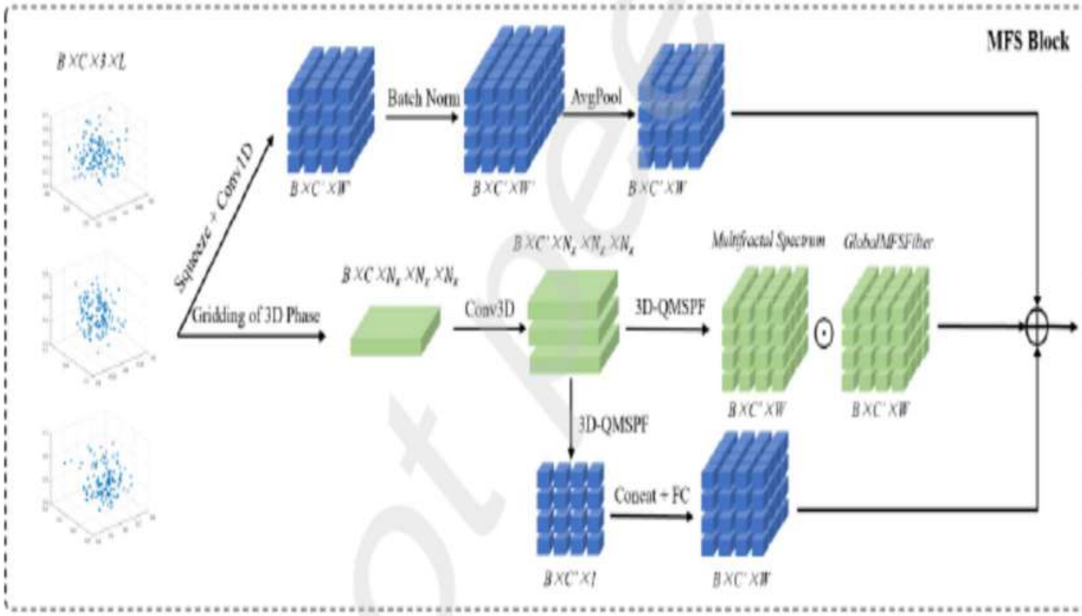
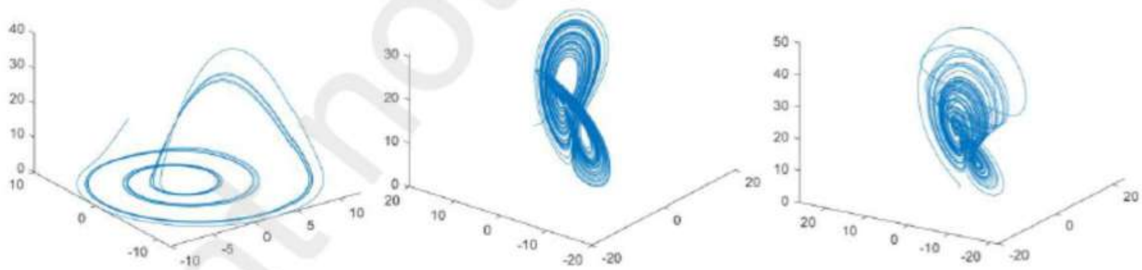


Figure 3. The network structure of the learnable MFS Module



(a) Rossler System (b) Lorenz System (c) Chen System

Figure 4. Phase Space Trajectories of Three Types of Chaotic Systems

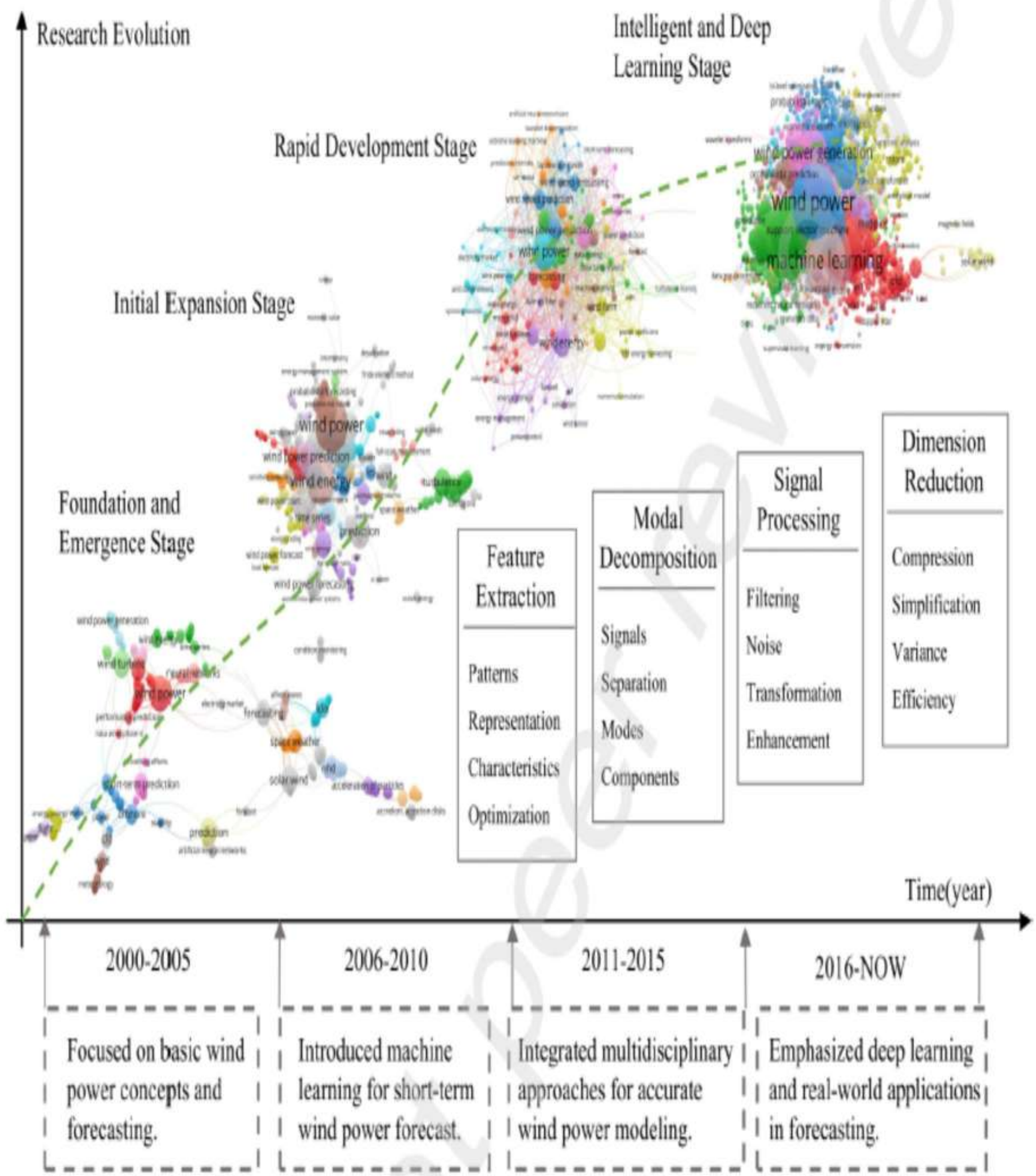


Fig.2 Evolution map of wind power forecasting research

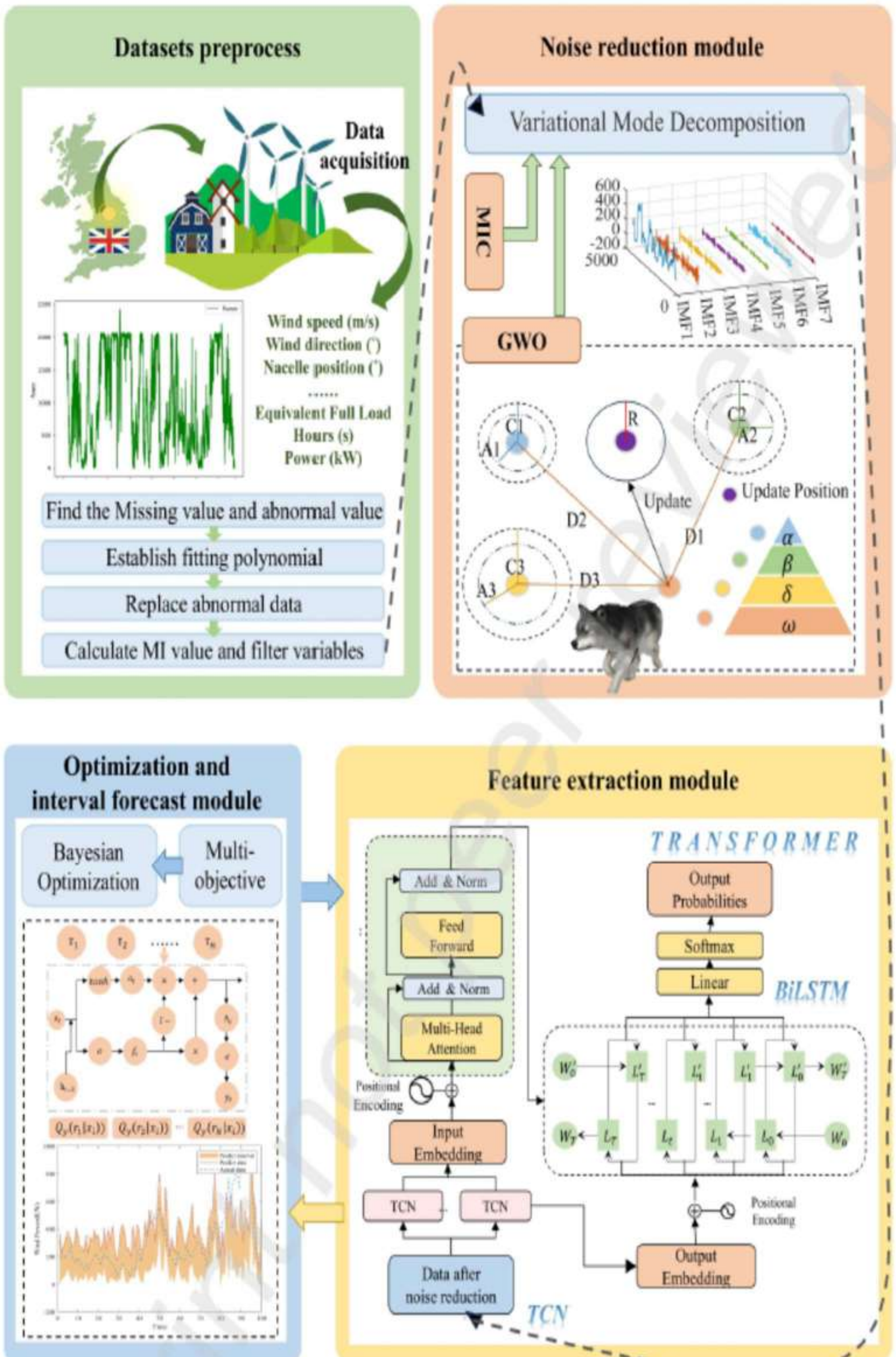


Fig.3 IVMD-MOBO-QRTBT model framework

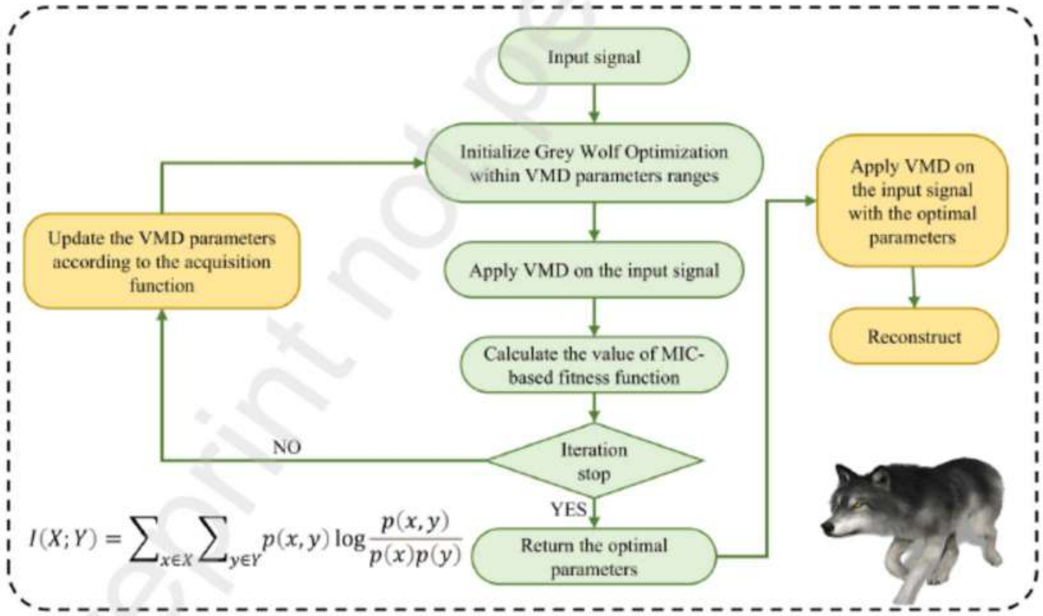


Fig.4 The process of noise reduction module

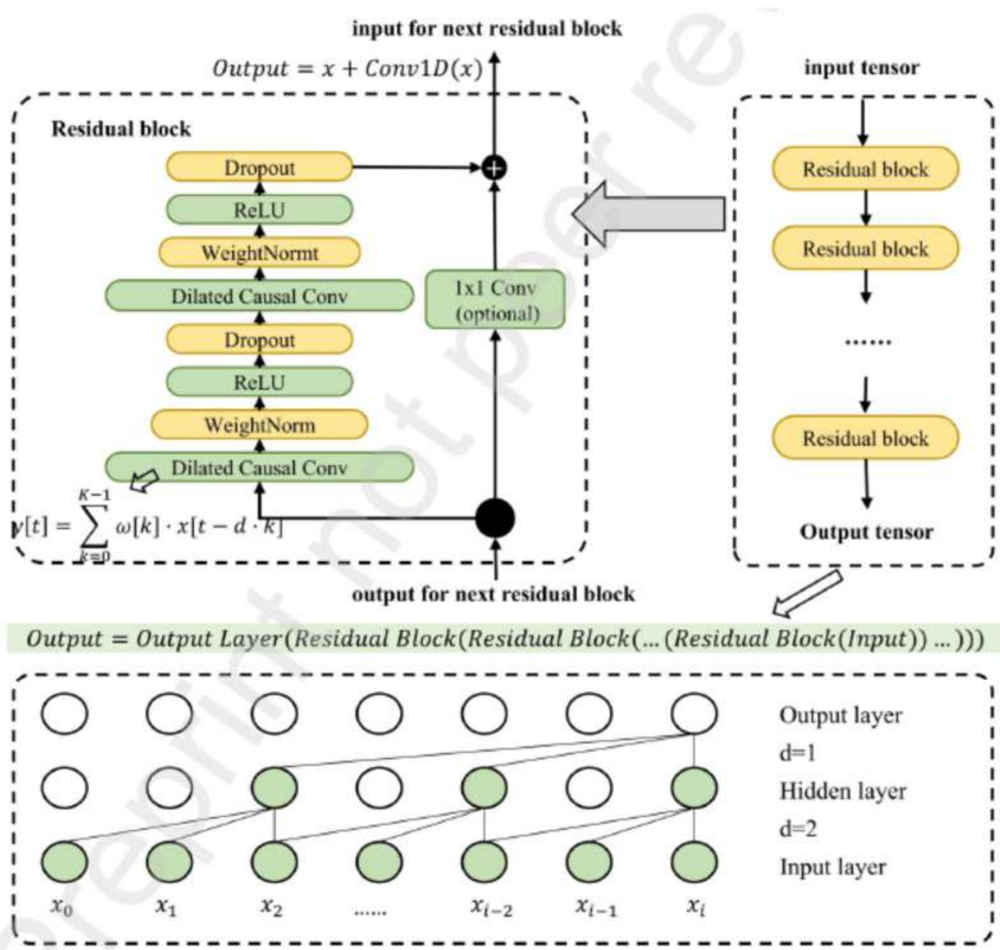


Fig.5 TCN network structure

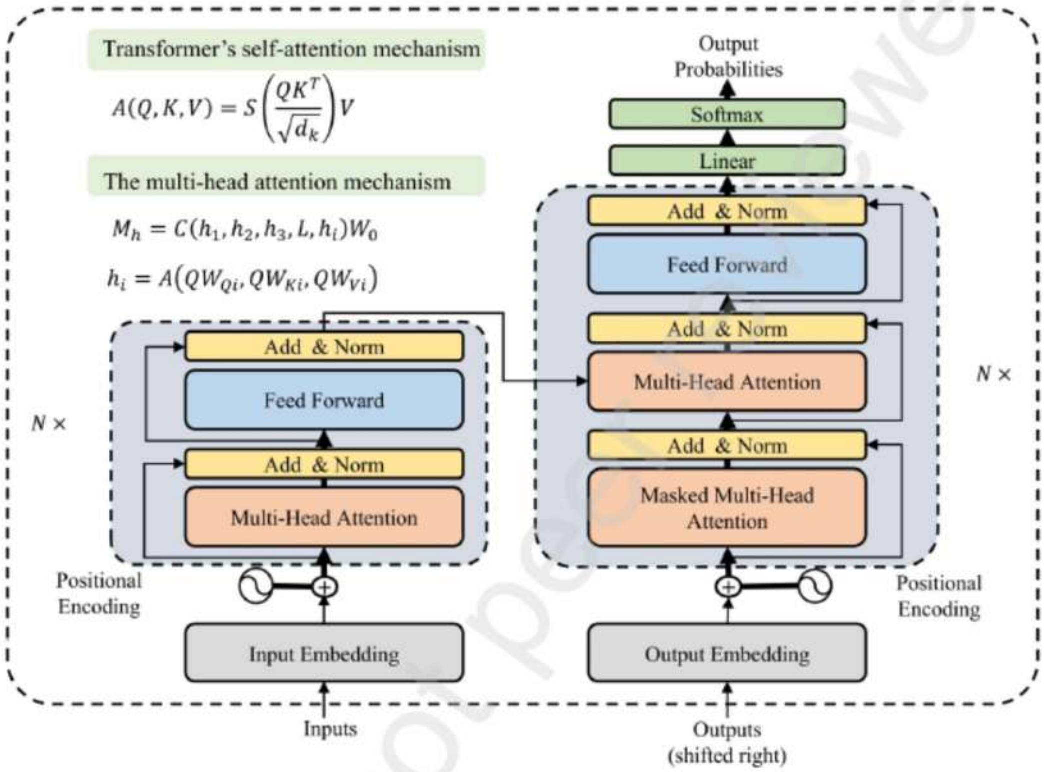


Fig.6 Transformer network structure diagram

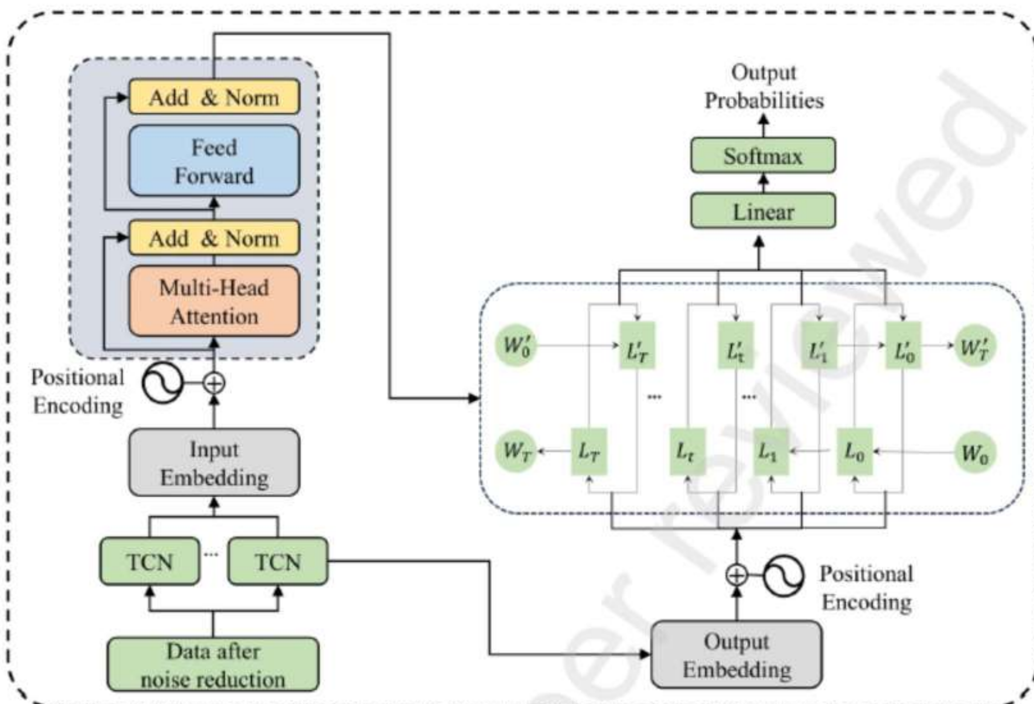


Fig.7 The structure diagram of feature extraction module

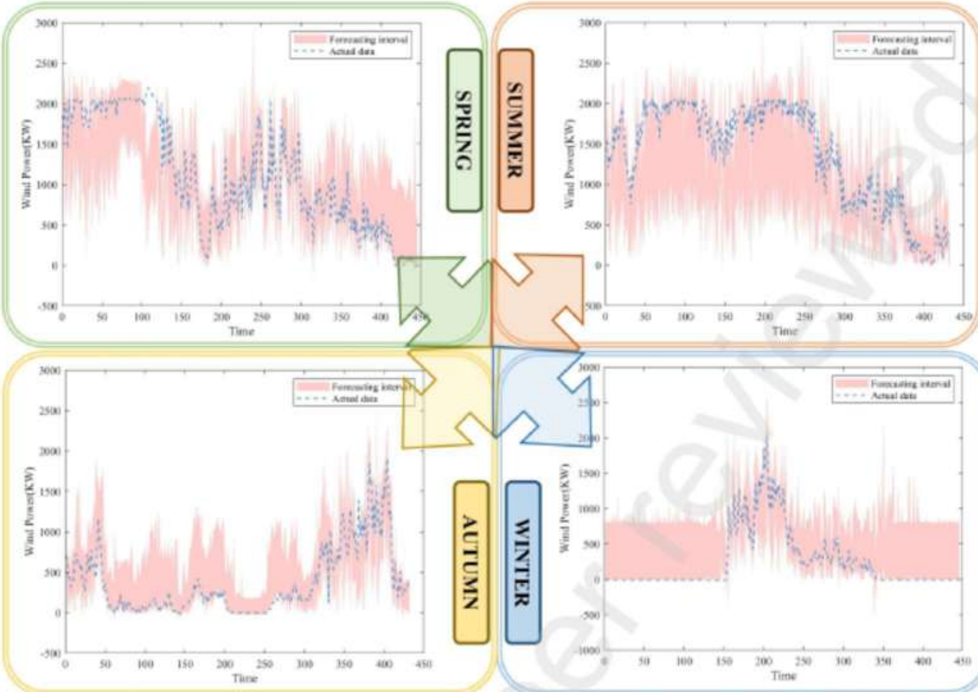
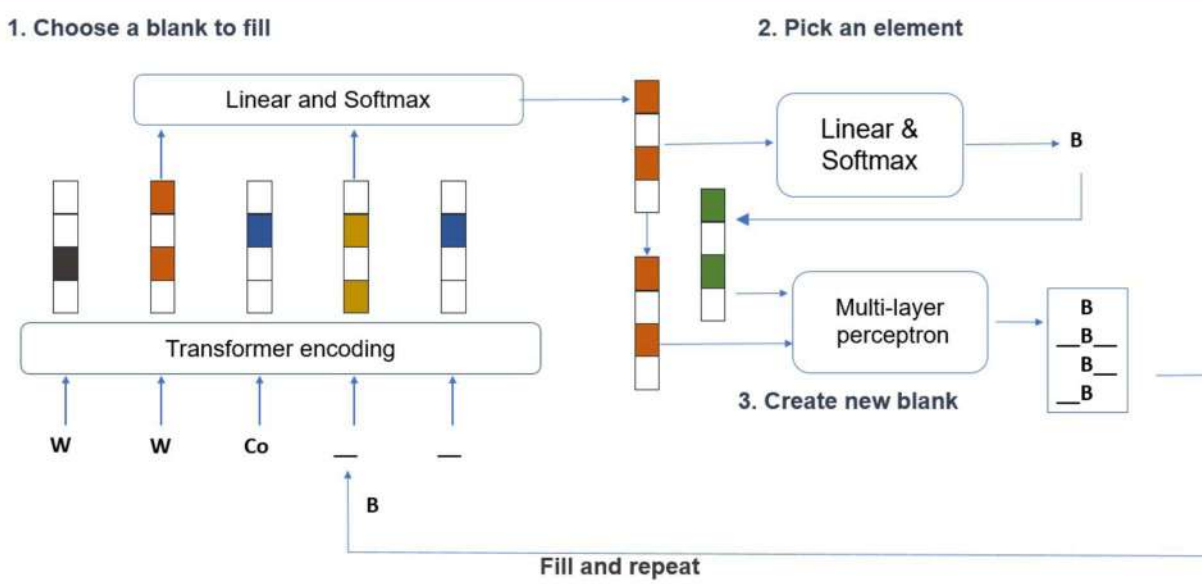


Fig.11 Interval forecasting by IVMD-MOBO-QRTBT model

Transformer Net	2025-19	



Architecture.blank-filling language model for material tinkering using W2CoB2

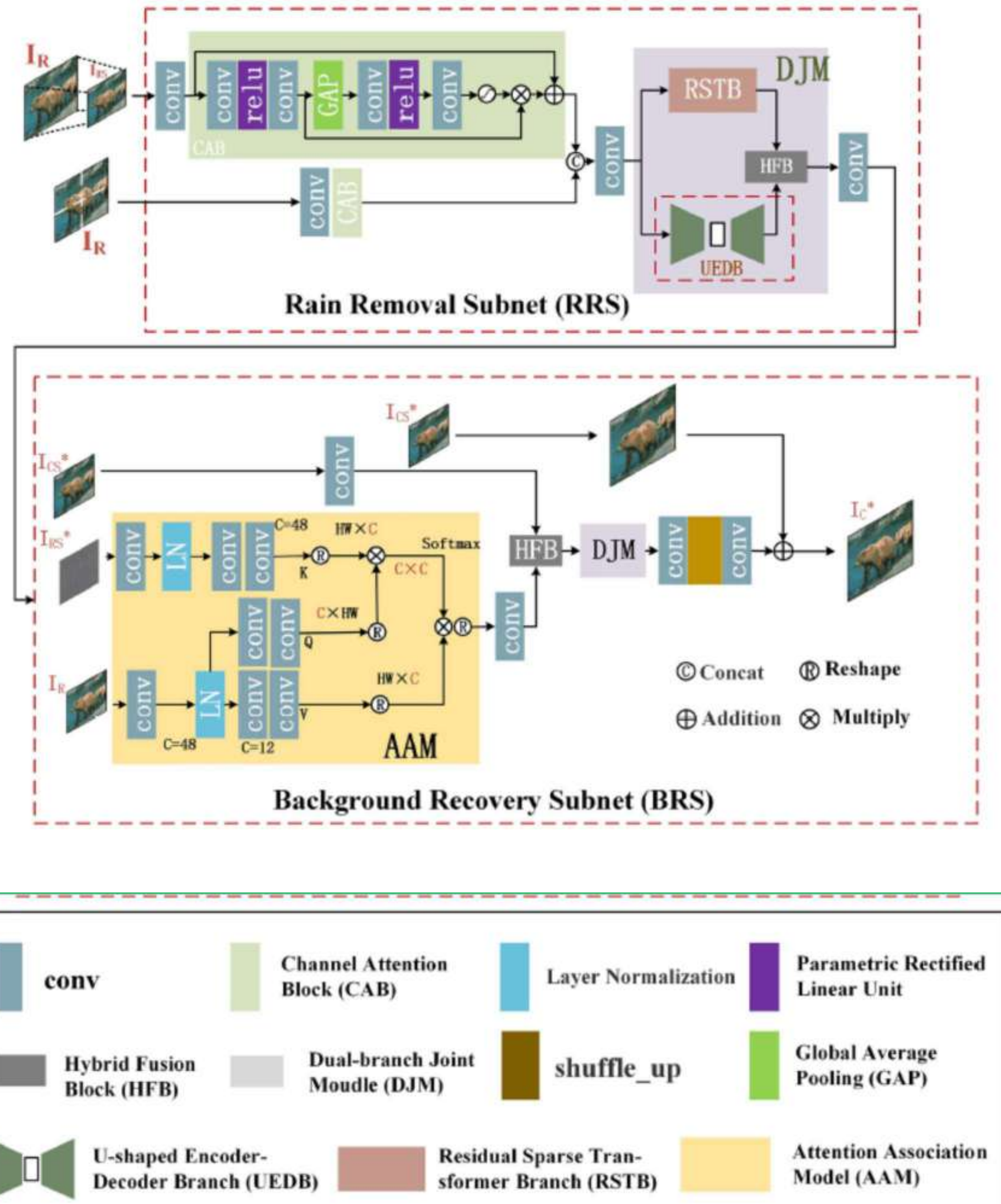
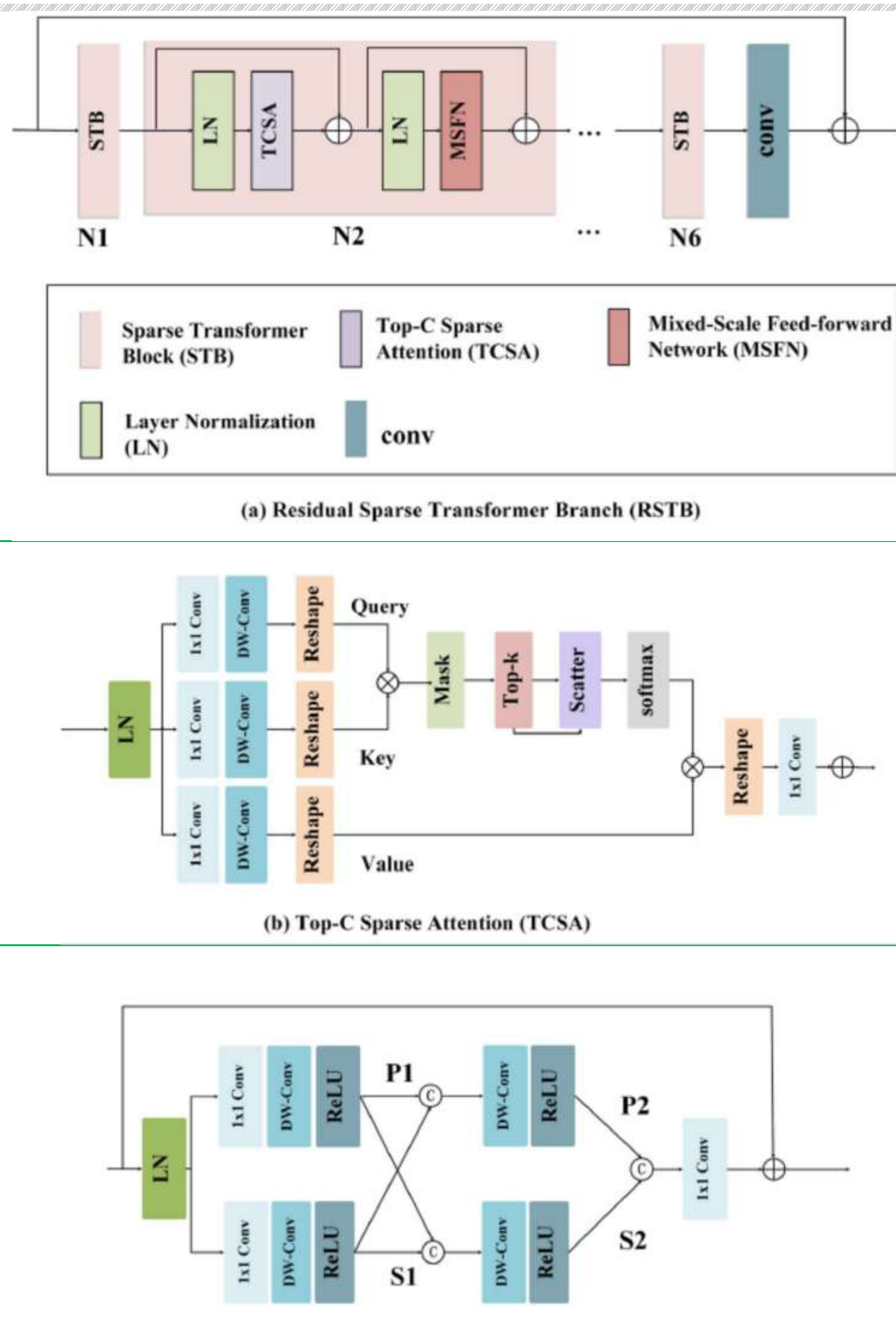


Fig. 1 Proposed network structure



(a) Residual Sparse Transformer Branch (RSTB)

(b) Top-C Sparse Attention (TCSA)

(c) Mixed-scale Feed-forward Network (MSFN)

Overview of the designed RSTB. a RSTB b TCSA c MSFN

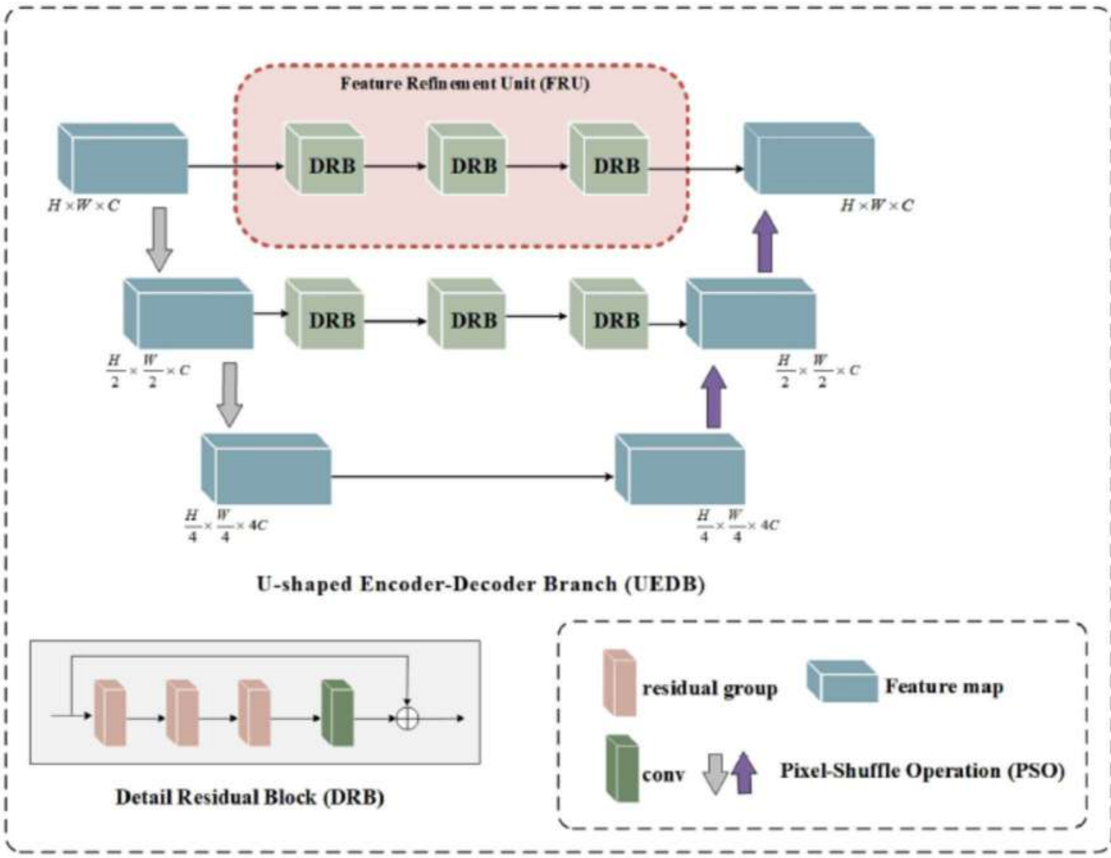
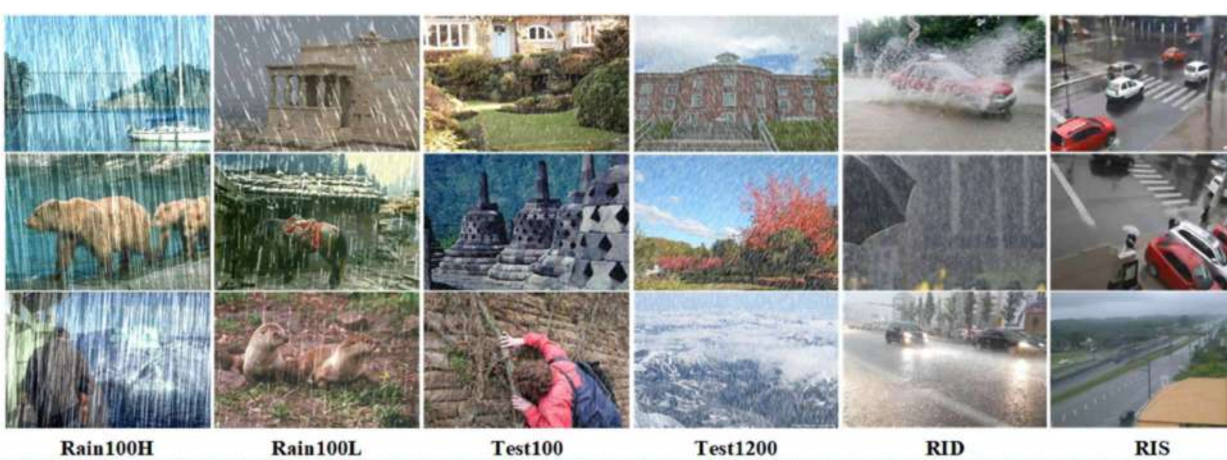


Fig. 3 UEDB, FRU, and DRB modules

Transformer Net	2025-20	



Example images from the datasets

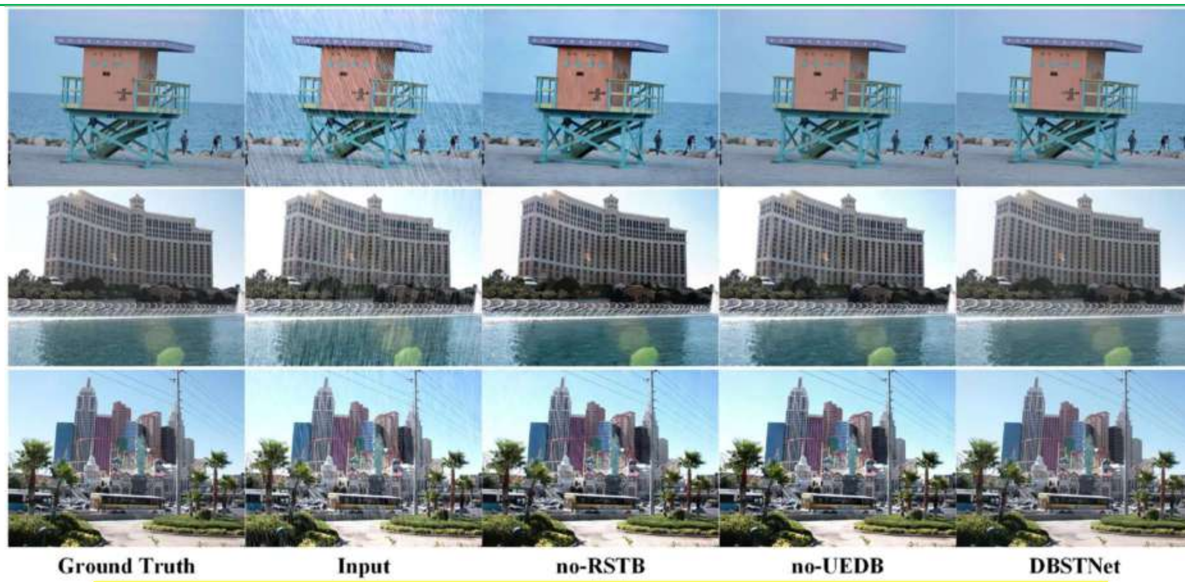
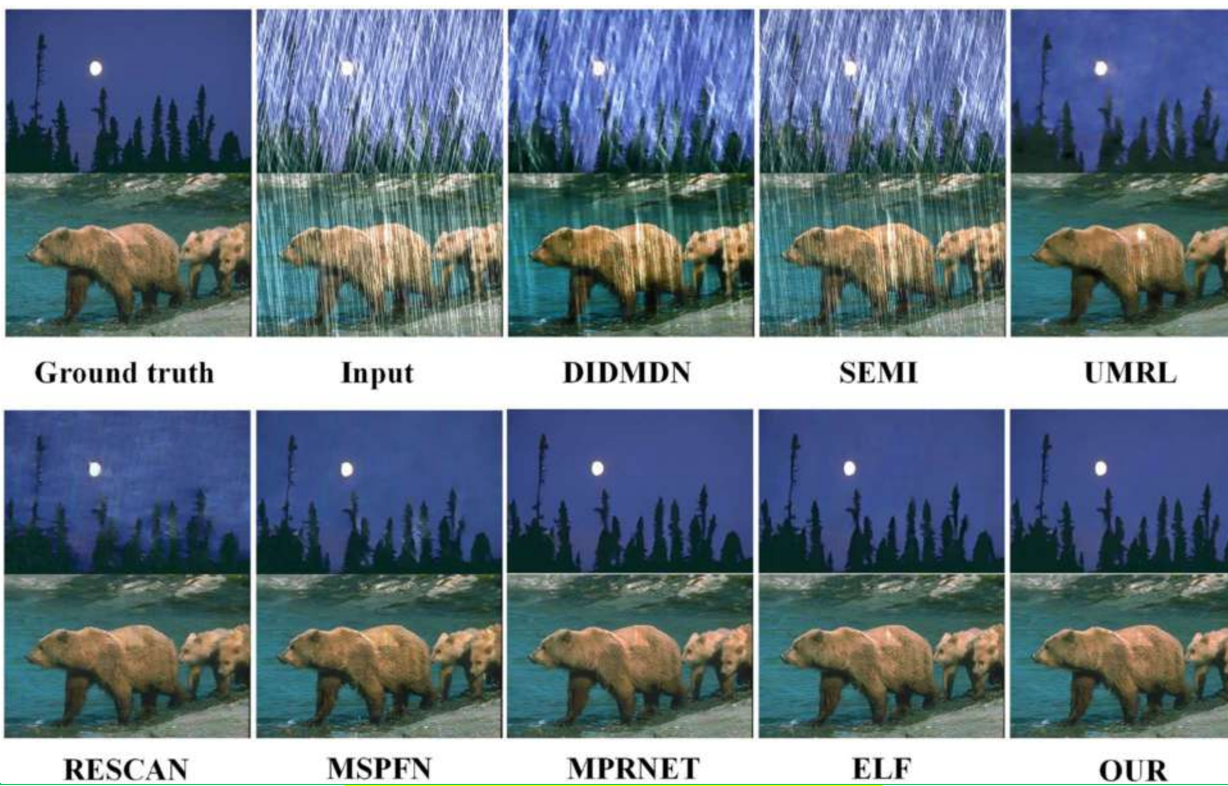
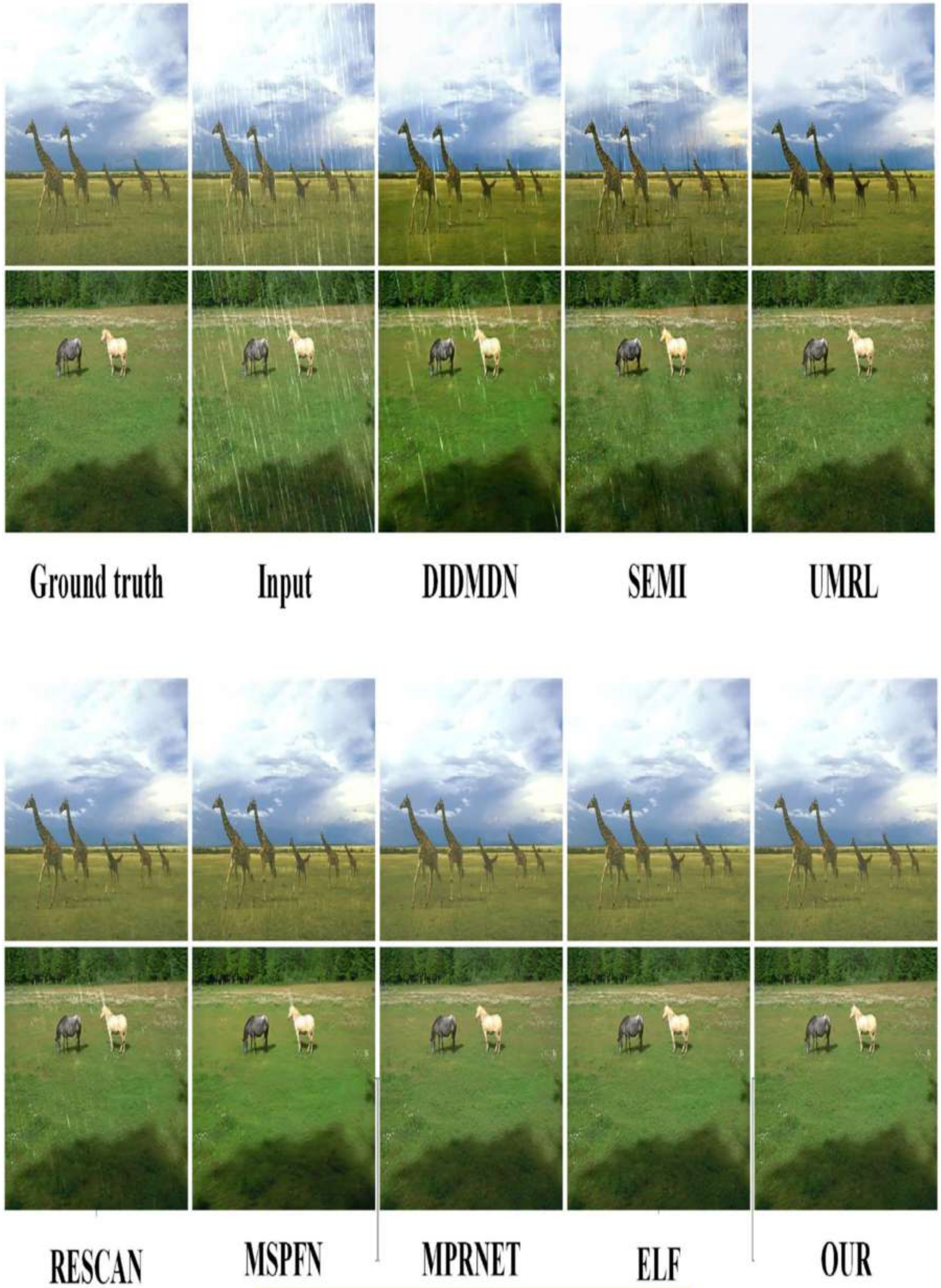


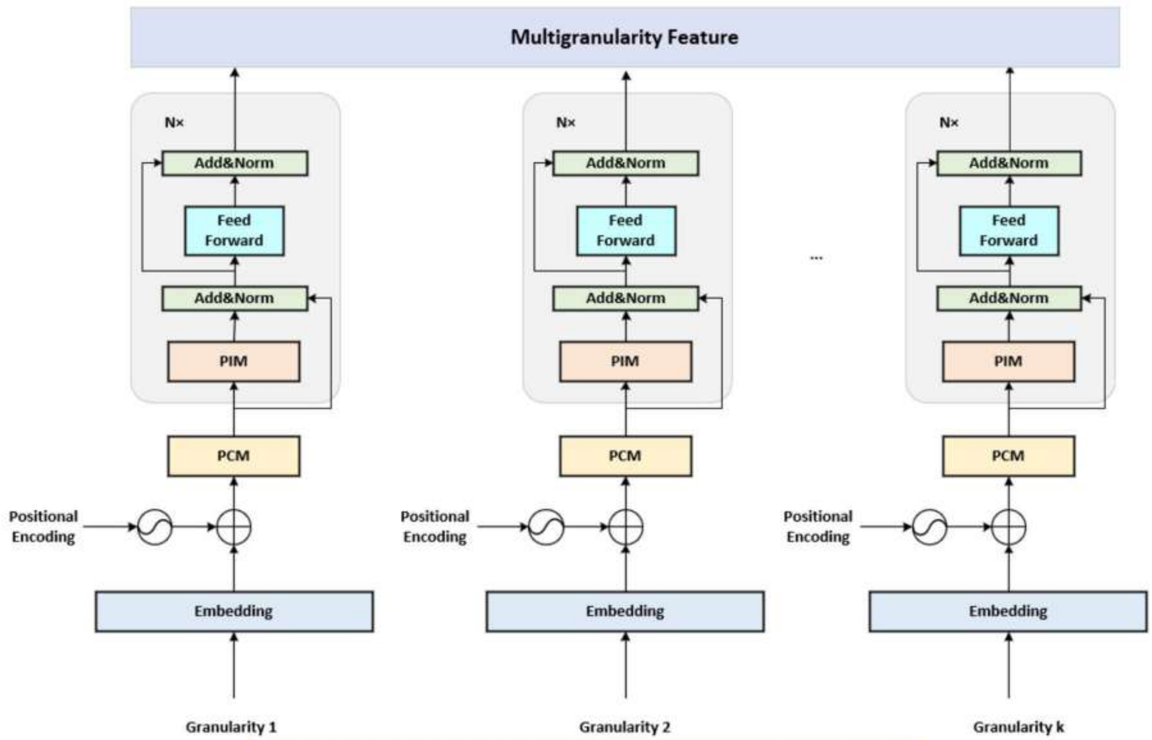
Fig. 5 Rain removal results of different branches in the dual-branch joint module



Deraining results on the Rain100H Dataset

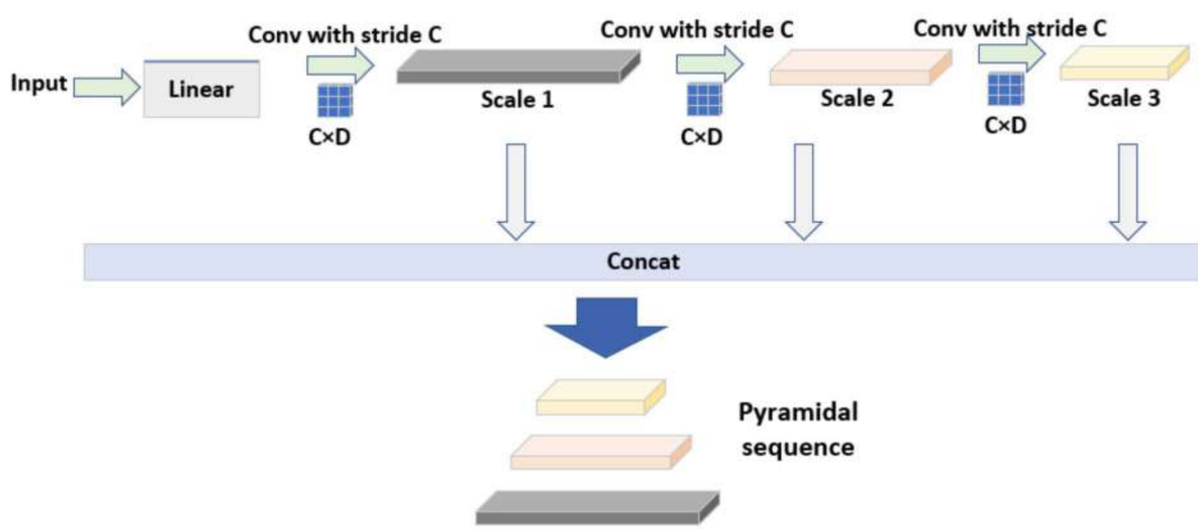


Deraining results on the Rain100L Dataset



Structure of parallel pyramidal Transformer module

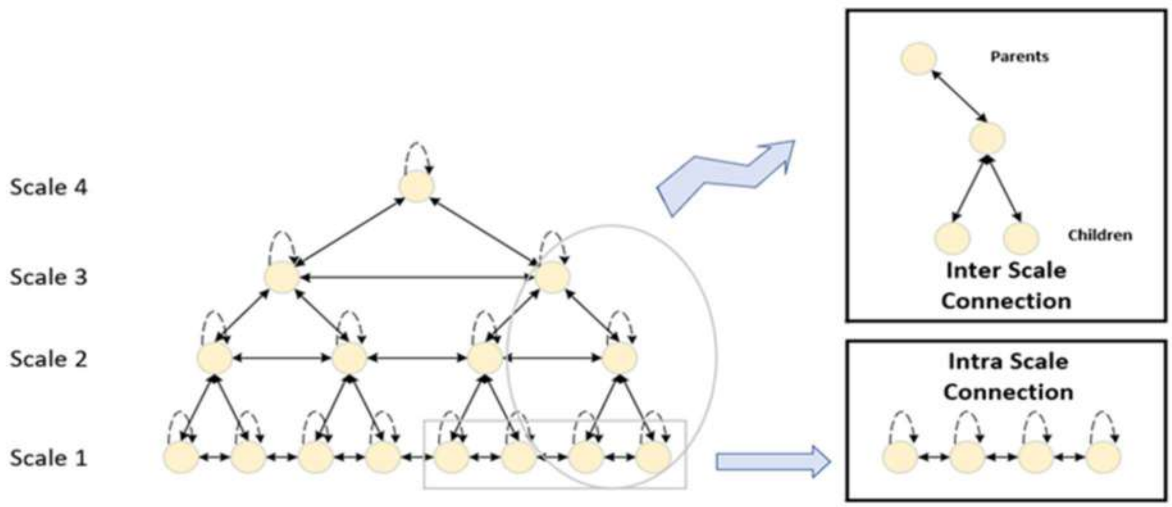
- ✓ PCM module uses multiple convolution operations to construct the input data into a pyramid structure, generating multilayer sequences from coarse to fine granularity.
- ✓ PIM module connects adjacent nodes at the same scale and parent–child nodes across different scales to capture the long-term and short-term dependencies of the time series.



Structure of pyramid construction module

Input data are first dimensionally reduced through a linear layer,

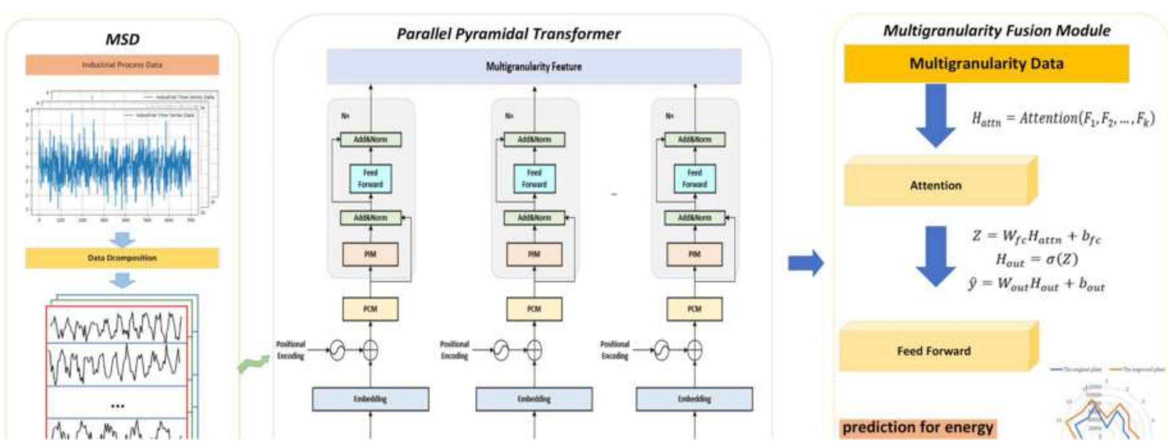
then processed through multiple convolutional layers to obtain sequences at different scales
Sequences at different scales are concatenated to form a pyramid structure.



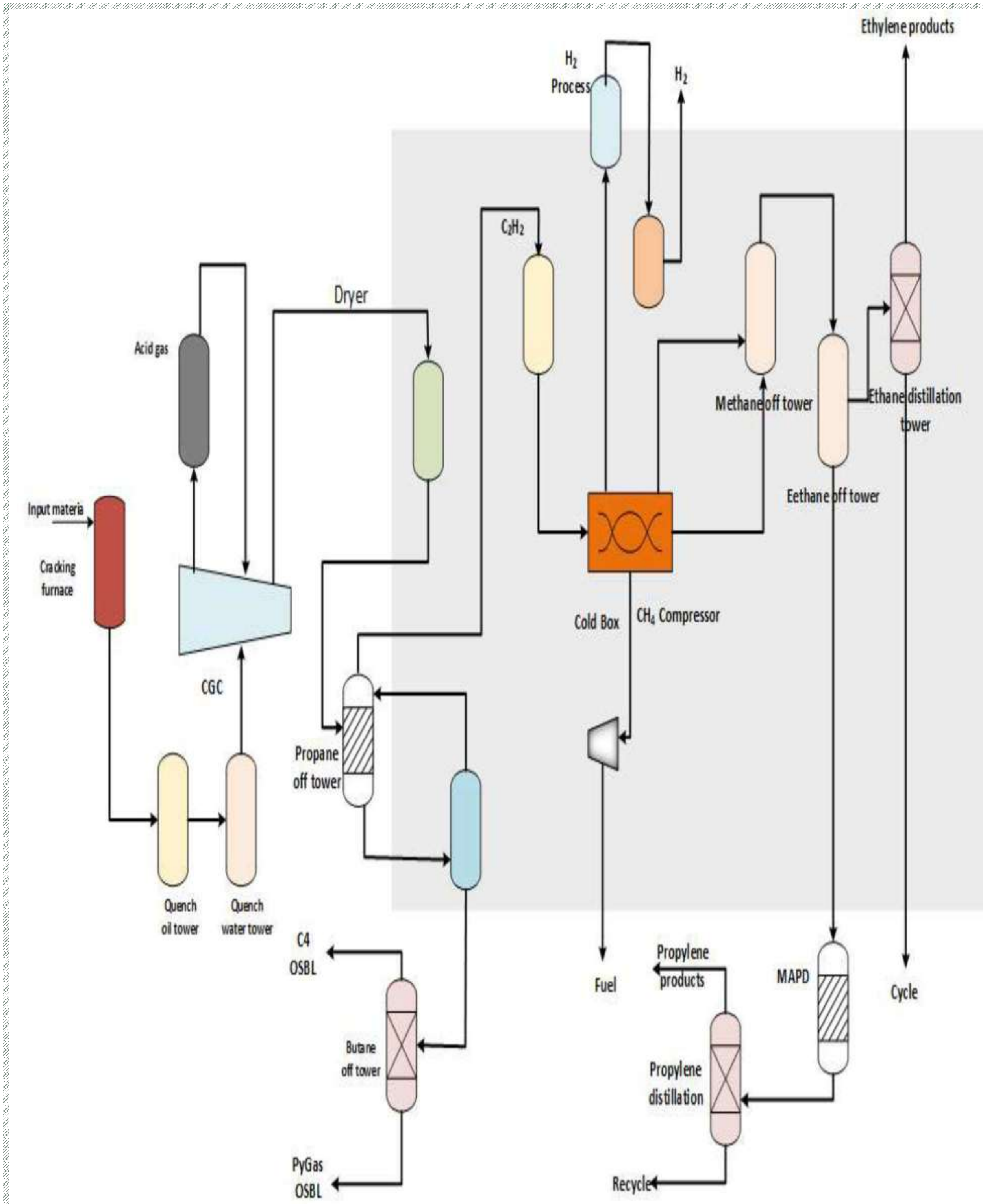
Structure of pyramid interaction module

The PIM consists of two parts: Intra-Scale connections and Inter-Scale connections. Inter-Scale connections construct a multi-resolution representation of the sequence and Intra-Scale connections capture temporal dependencies at different resolutions by connecting consecutive neighboring nodes

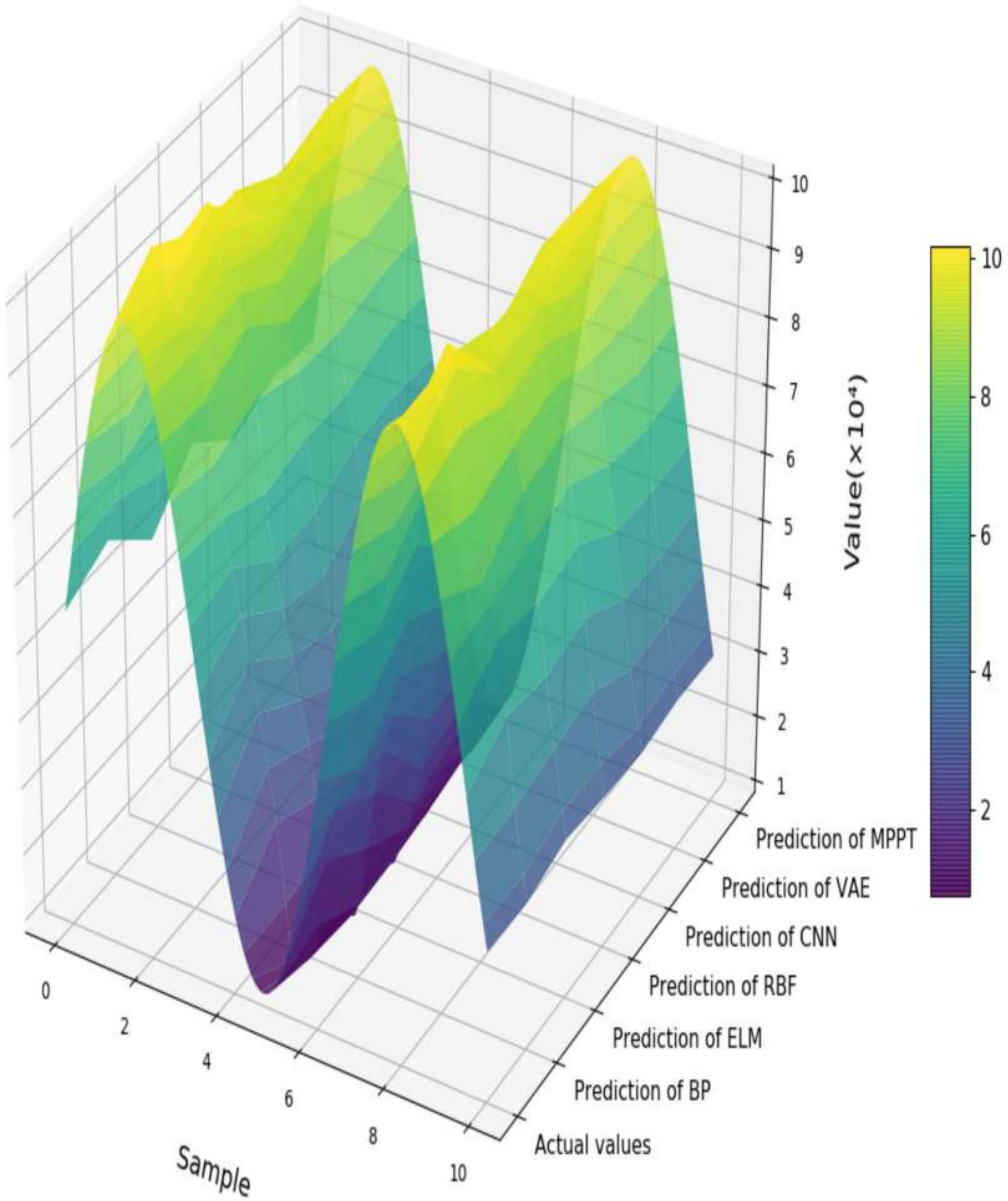
Transformer Net	**2025-20**	



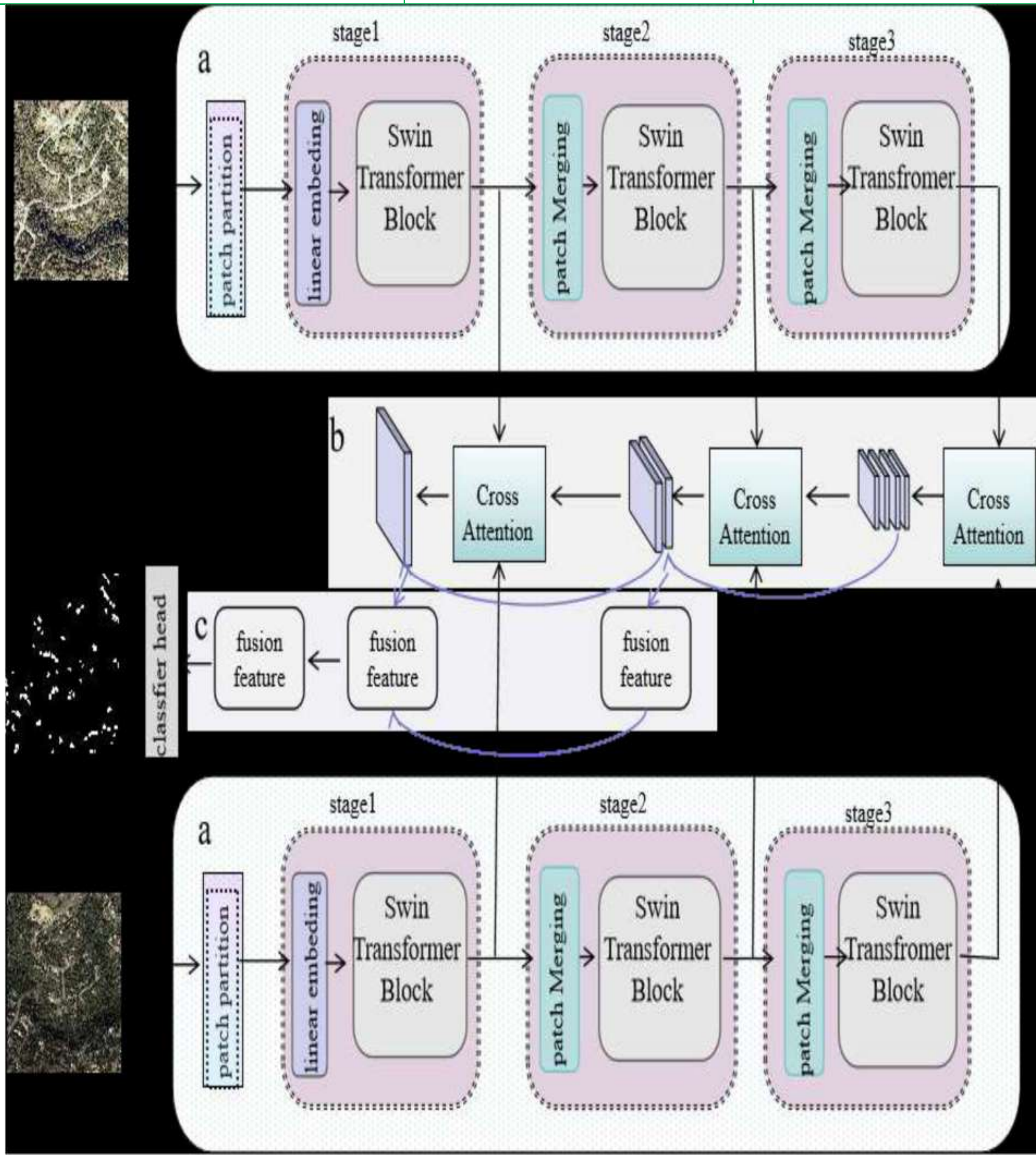
Overview of multigranularity parallel pyramidal Transformer which consists of MSD, PPT, and FM.



Production process of the ethylene plant

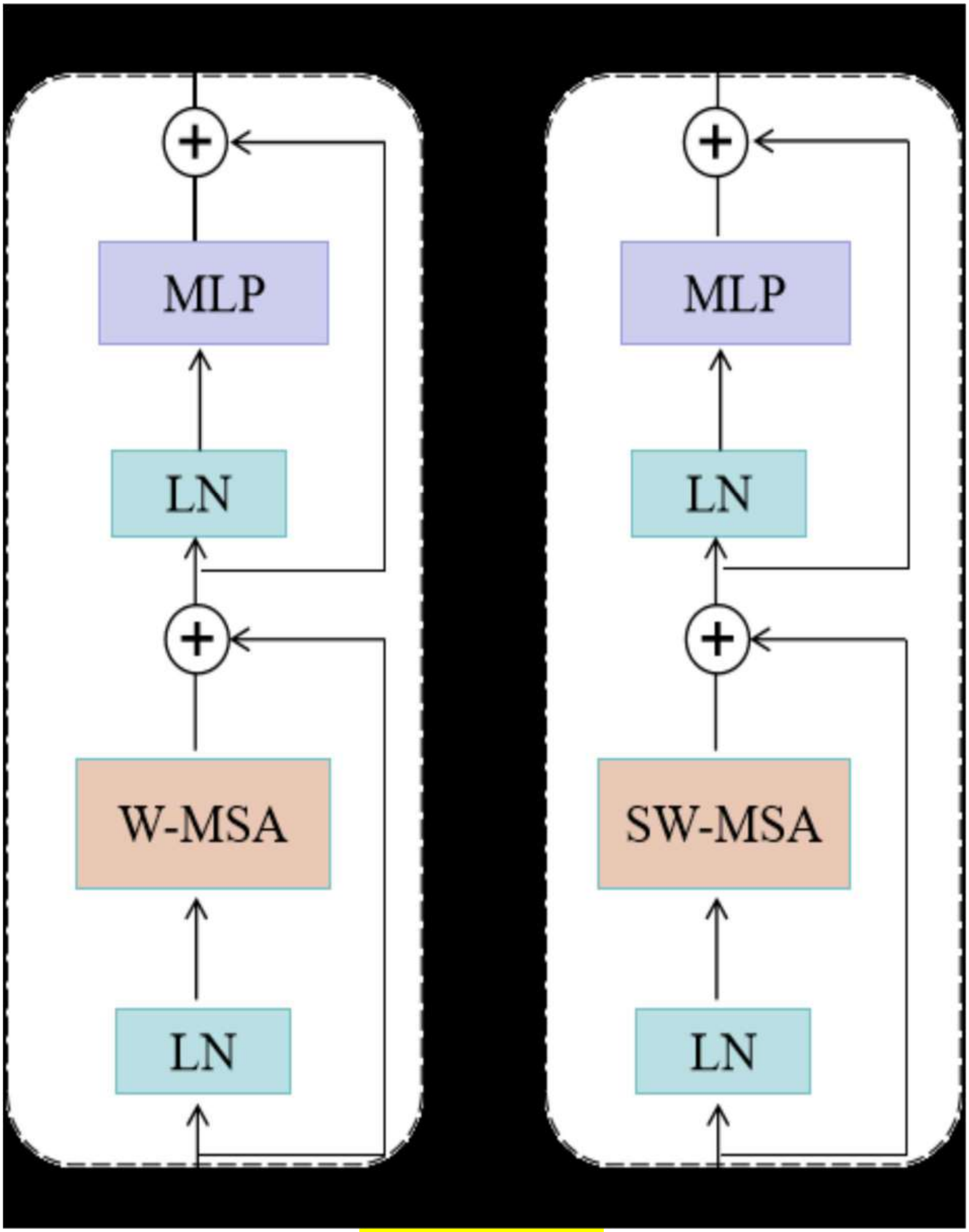


Comparison of prediction results

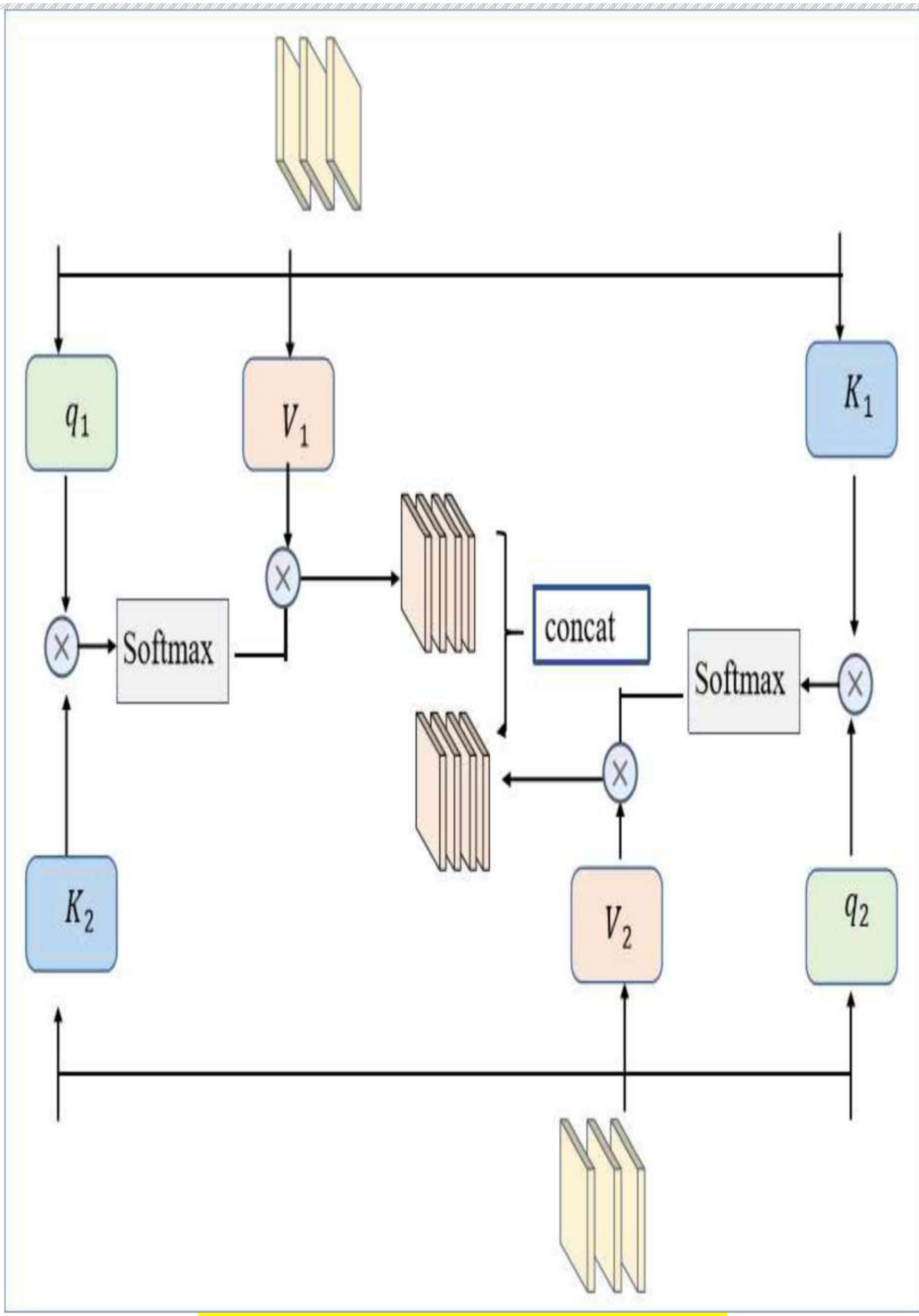


Overall structure of the SCACD-Net.

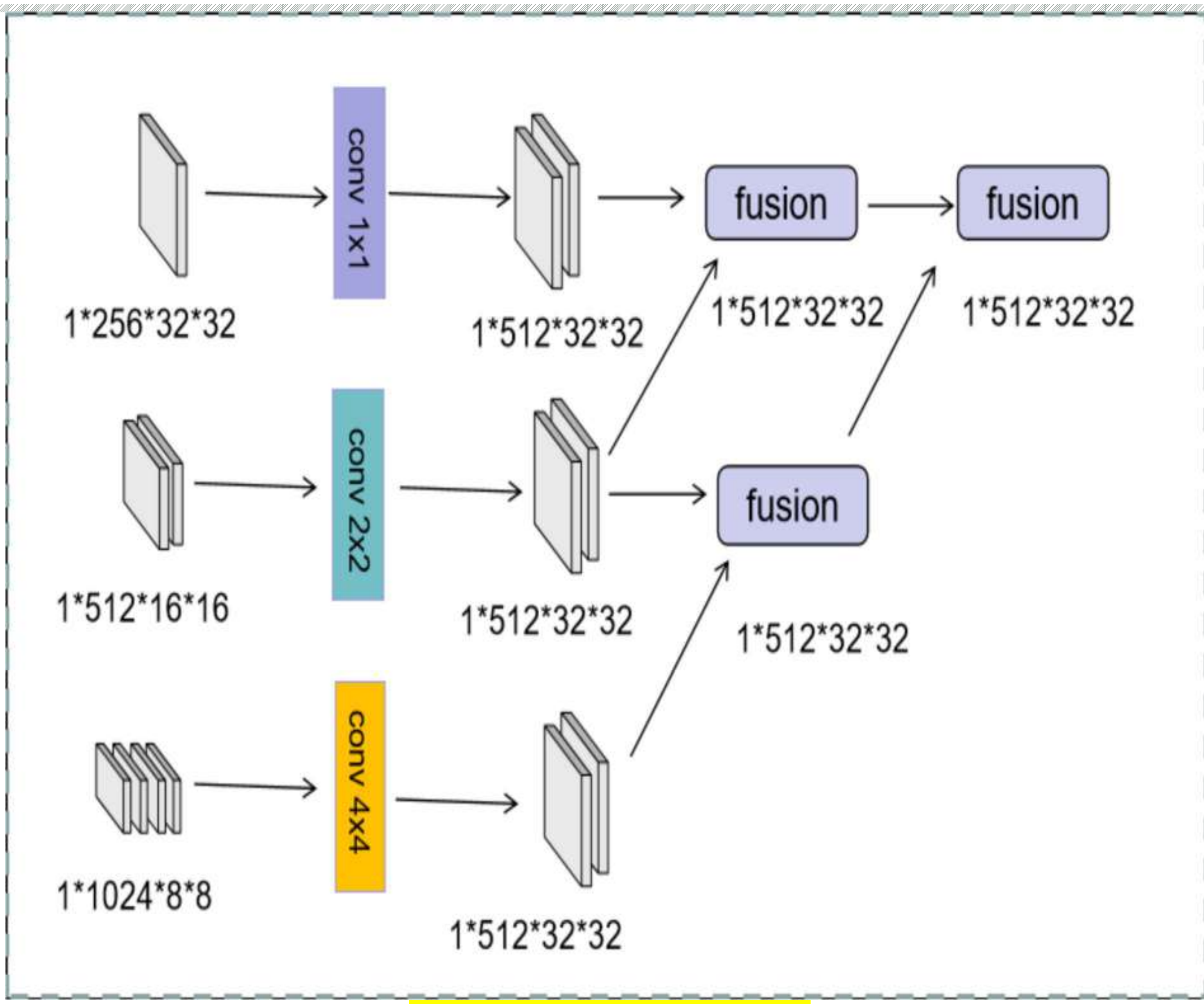
(a) Feature extraction network
(b) Cross-attention difference feature extraction network
(c) Cascaded feature fusion network



Swin Transformer block.



Cross-Attention difference Feature Extraction Module.



Cascade feature fusion network.

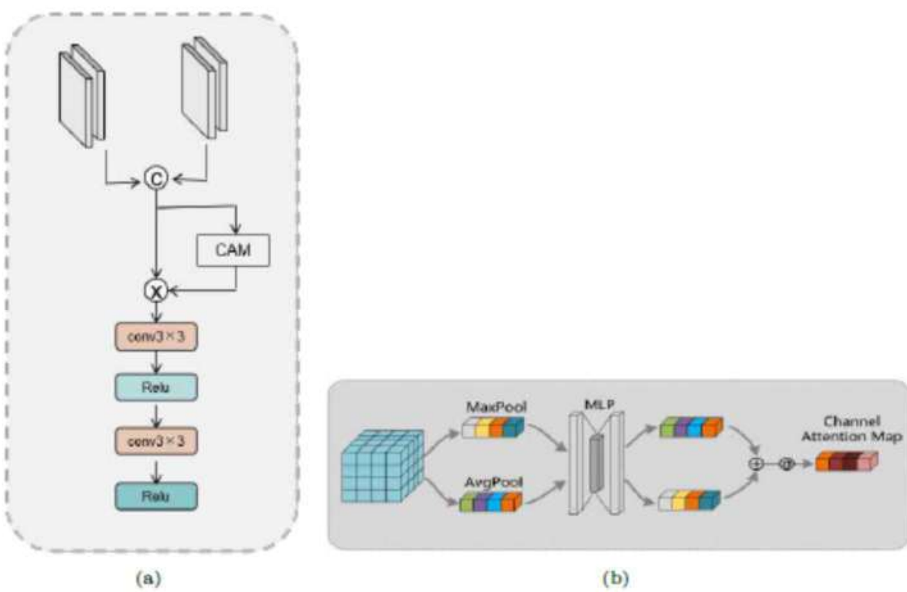


Fig. 5 (a) Feature Fusion block. (b) Channel Attention Mechanism (CAM)

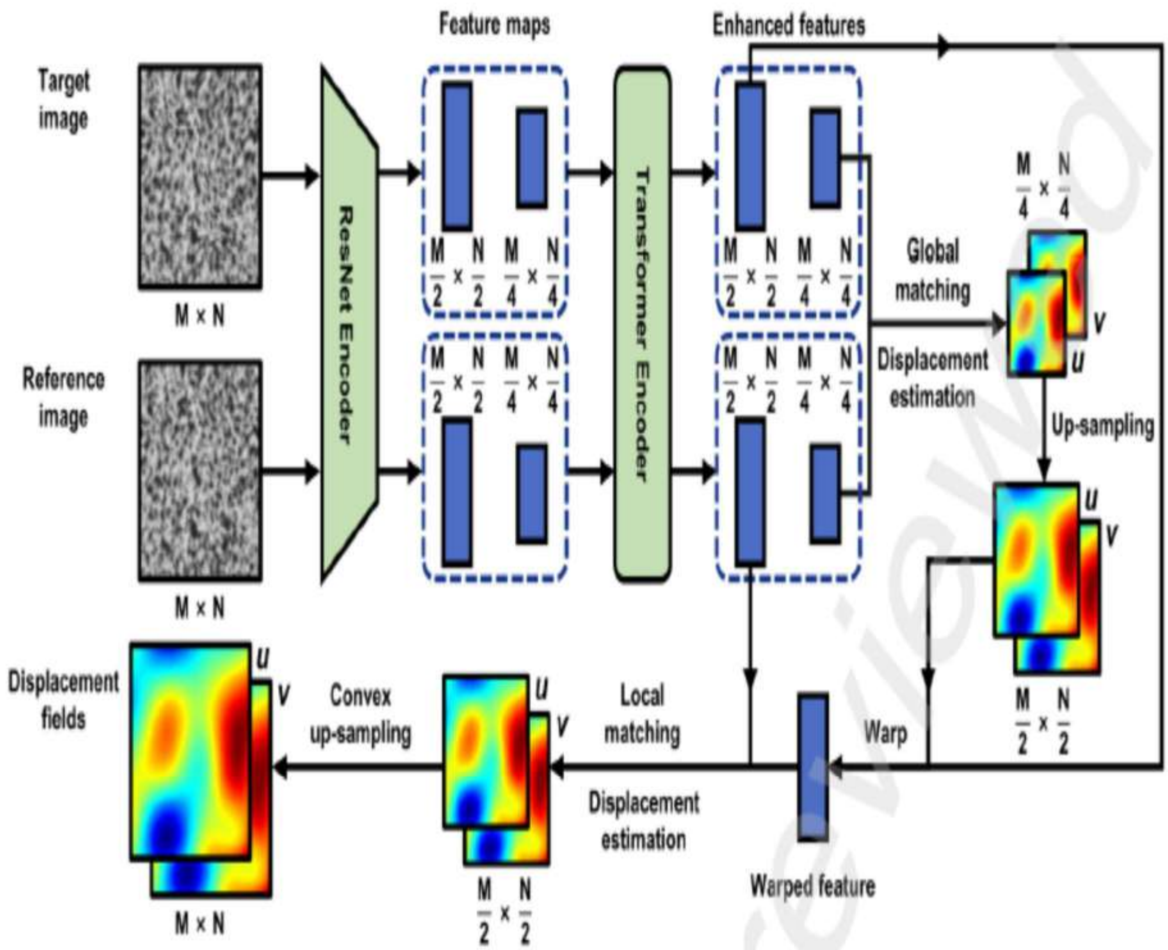


Figure 1. Architecture of DICTr.

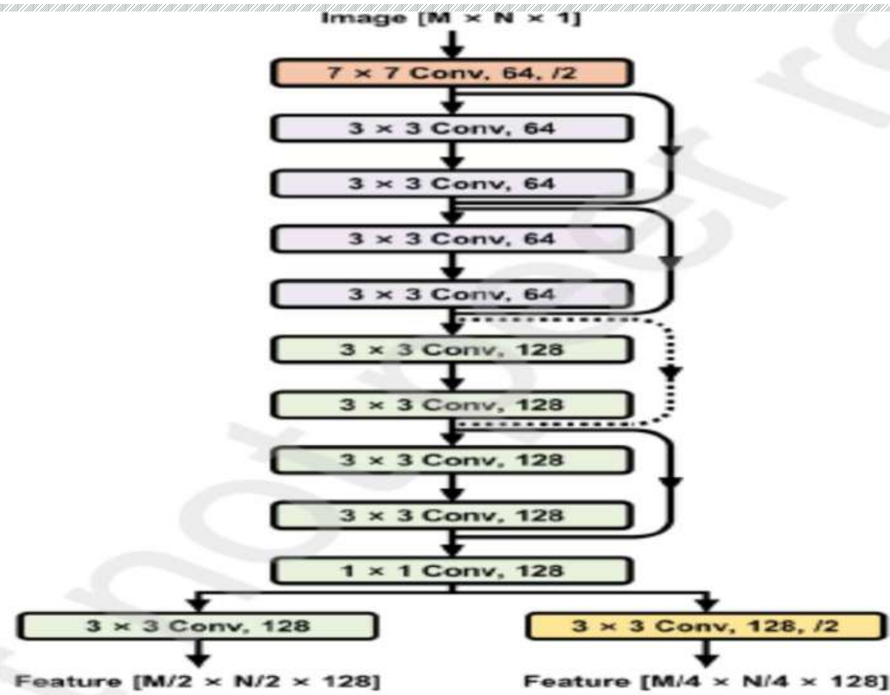


Figure 2. ResNet network for feature extraction. Additional connections represent the skip connections (also called shortcuts) of the residual network. The dashed shortcut represents the operations to increase the channels of the input.

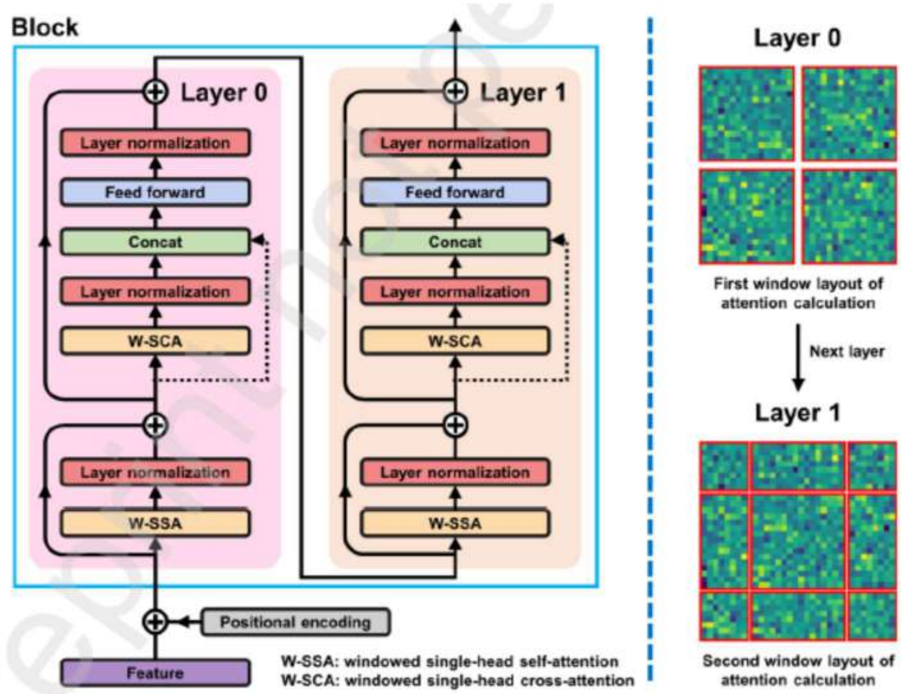
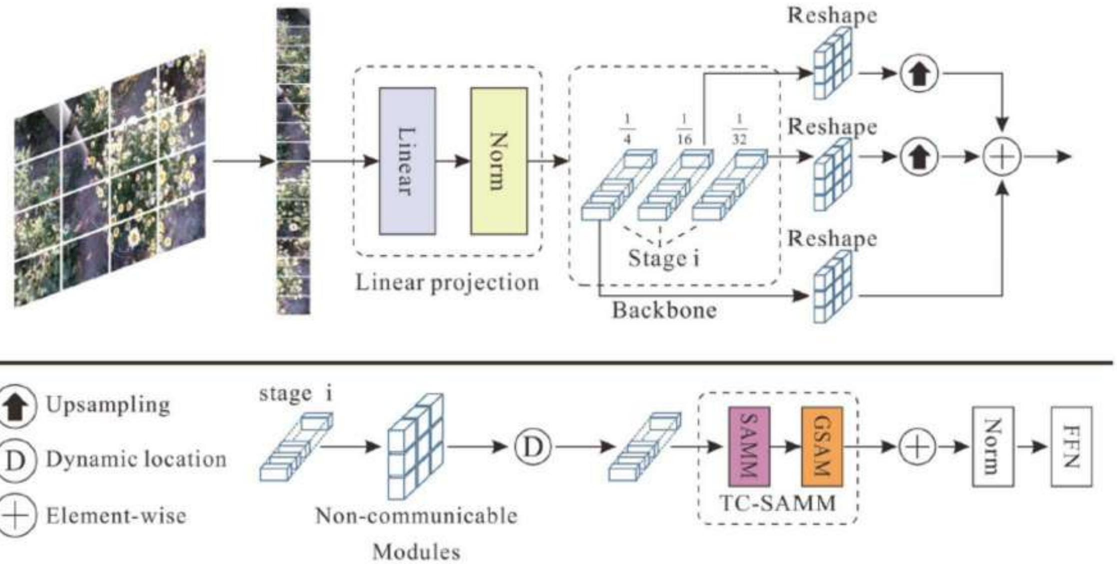


Figure 3. Transformer-based block for feature enhancement (left) and two different window layouts for the attention calculation in Layer 0 and Layer 1 (right). Dashed shortcuts represent concatenation of the input and the output.

Transformer Net	2025-24	



Structure of the proposed TCViT (tea chrysanthemums Vision Tr)

- ✓ Between the generated predicted density map and the ground truth of the density map, an improved loss function and back propagation are used for the final exact counts

Transformer Net	2025-25	

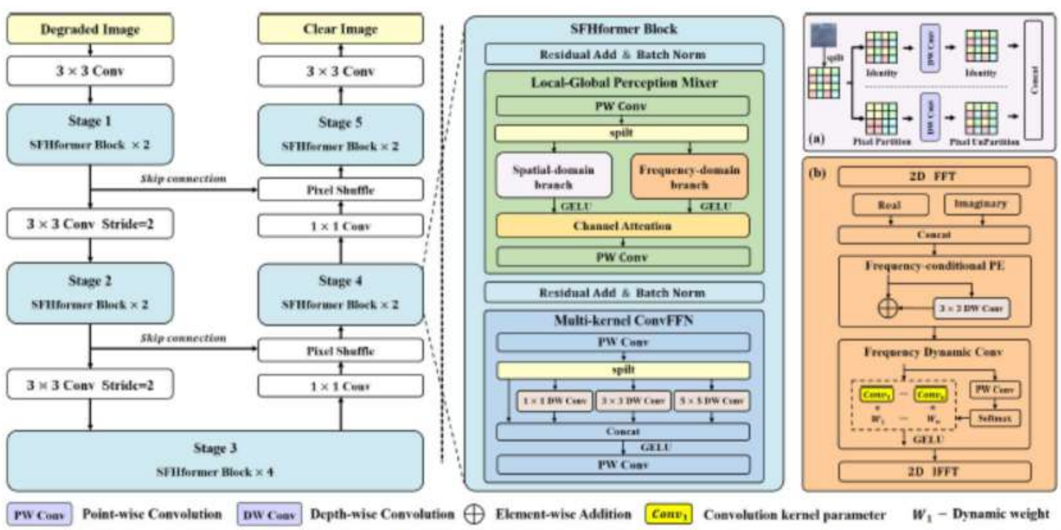


Fig. 2: Overview of the proposed SFHformer. (a) details of spatial-domain branch and (b) details of frequency-domain branch.

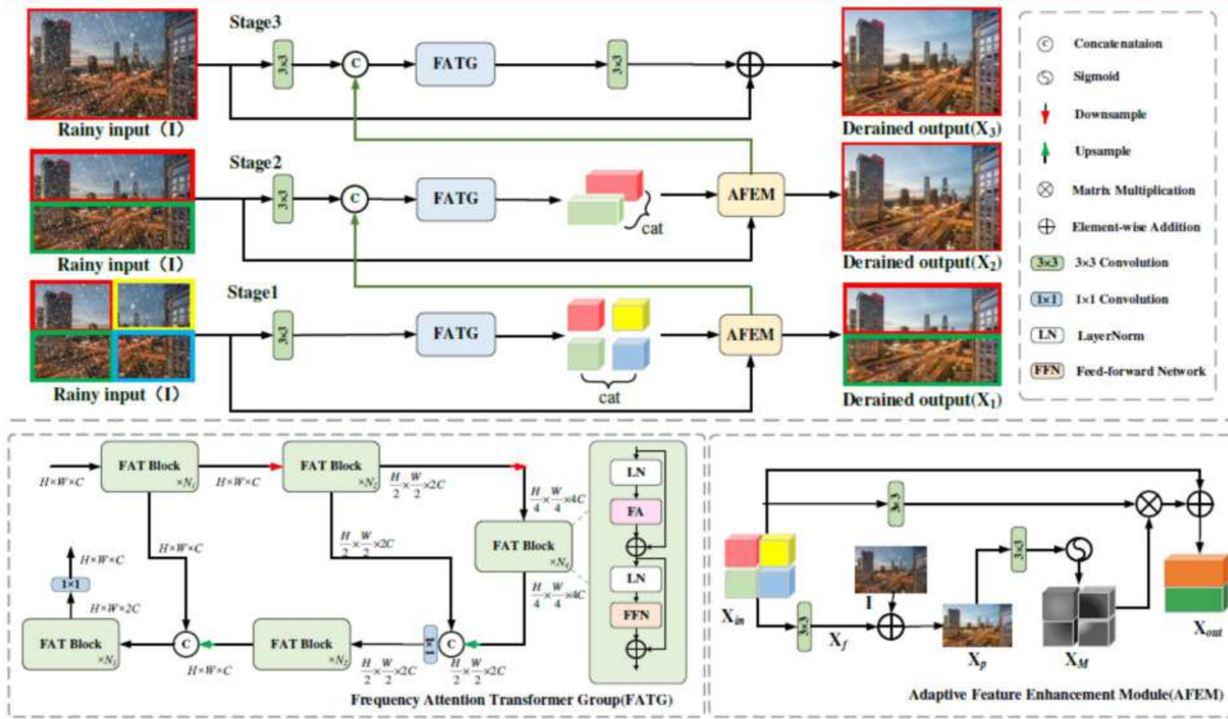
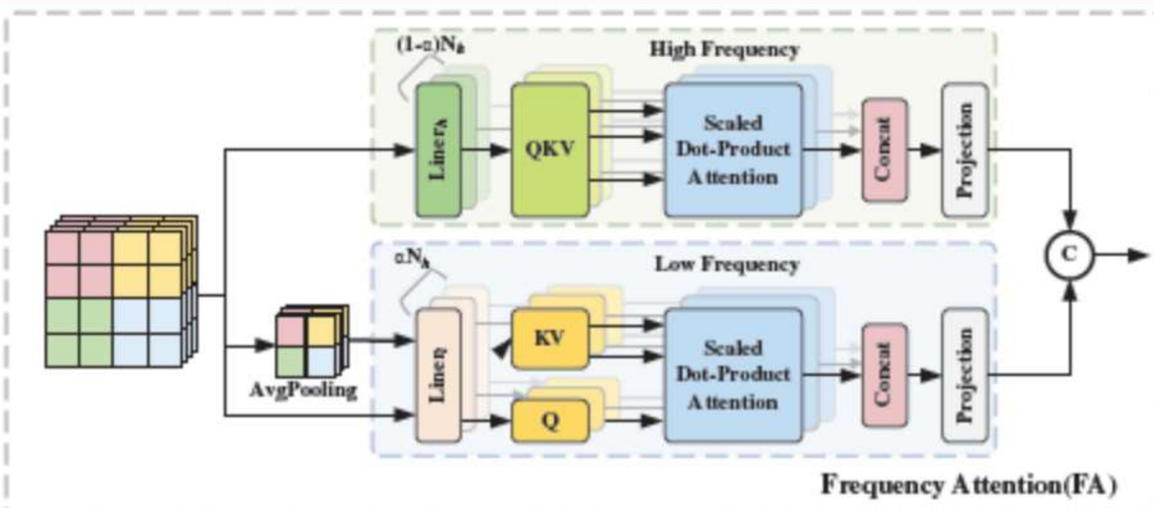


Fig. 1 The overall architecture of the proposed Multi-Patch De-raindrop Transformer for UAV images (MPDT), which mainly contains frequency attention Transformer block(FATB) and Adaptive Feature Enhancement Module(AFEM).

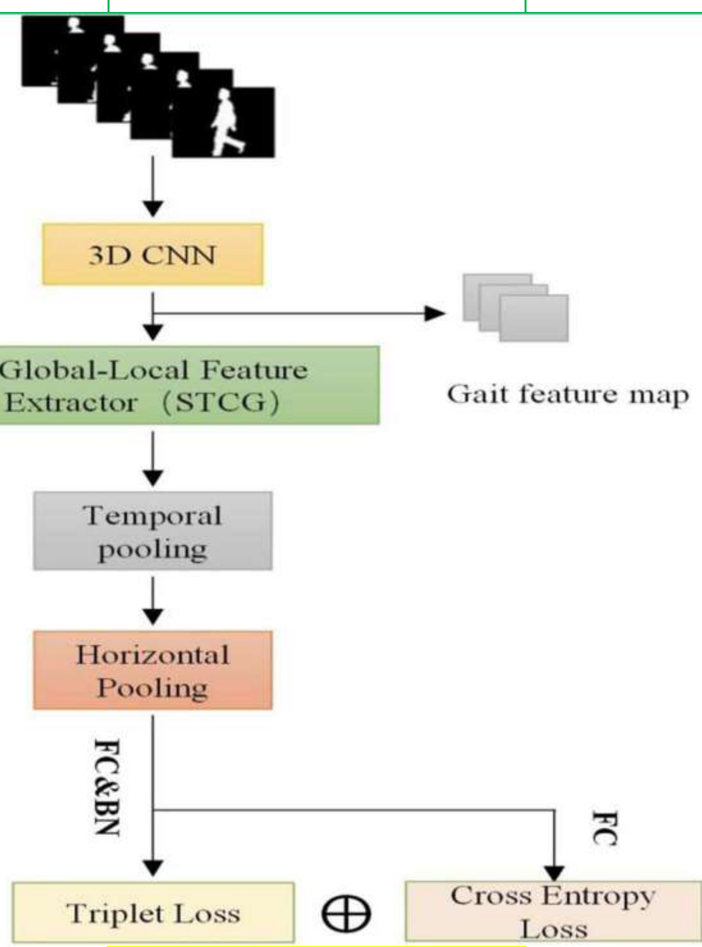


Frequency Attention mechanism.

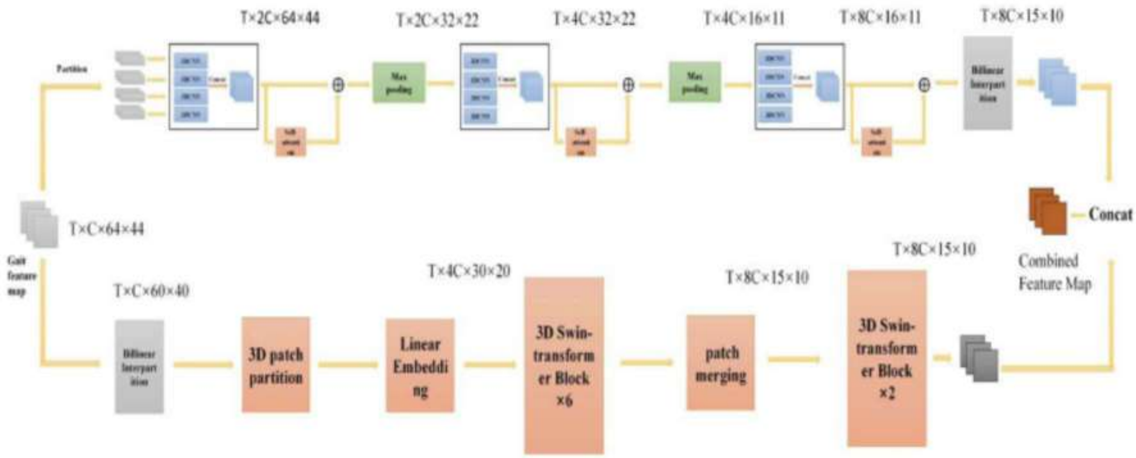
Table 2 Quantitative comparisons on Rain200H/L datasets for different methods.

Datasets		Rain200L		Rain200H	
Metrics		PSNR	SSIM	PSNR	SSIM
Prior-based methods	DSC [1]	27.16	0.8663	14.73	0.3815
	GMM [2]	28.66	0.8652	14.50	0.4164
CNN-based methods	PReNet [29]	37.80	0.9814	29.04	0.8991
	RCDNet [3]	39.17	0.9885	30.24	0.9048
	MPRNet [5]	39.47	0.9825	30.67	0.9110
	DualGCN [30]	40.73	0.9886	31.15	0.9125
	SPDNet [4]	40.50	0.9875	31.28	0.9207
Transformer-based methods	Uformer [31]	40.20	0.9860	30.80	0.9105
	Restormer [13]	40.99	0.9890	32.00	0.9329
	IDT [15]	40.74	0.9884	32.10	0.9344
	DRSformer [14]	41.23	0.9894	32.18	0.9330
	Ours	**41.54**	**0.9898**	**32.52**	**0.9349**

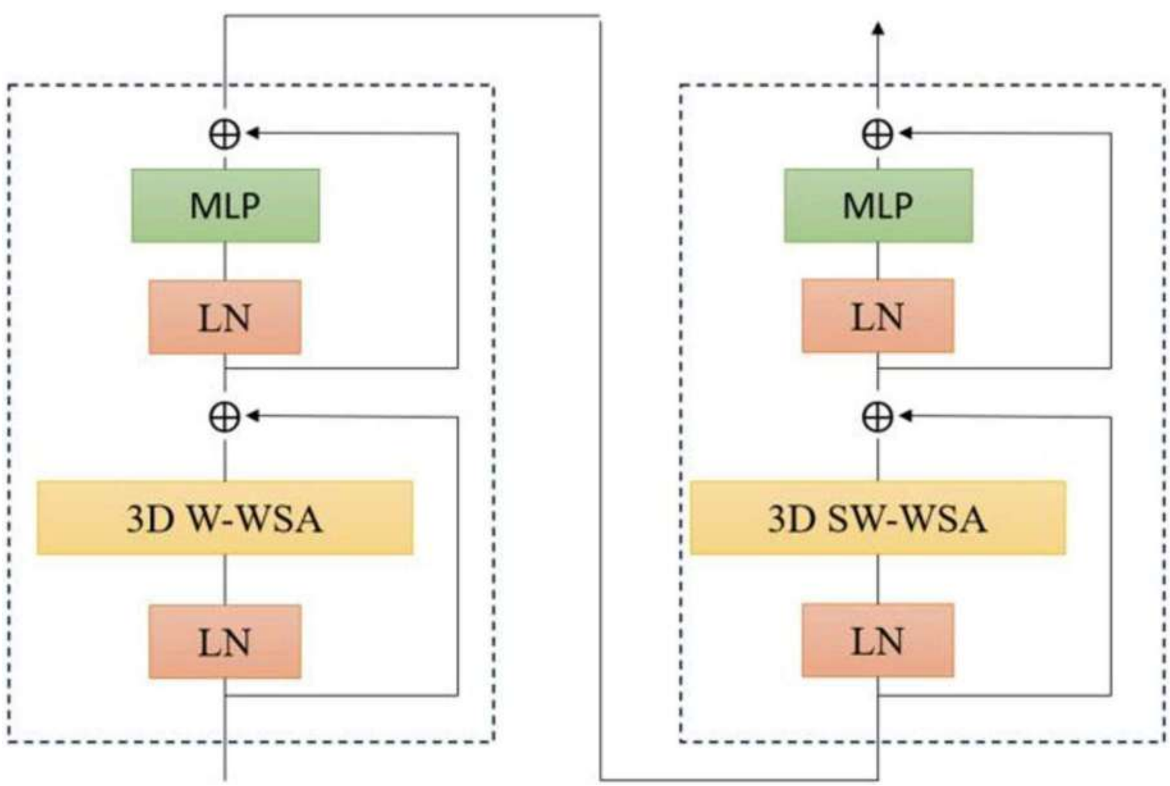
Transformer Net	2025-27	



Mainstream of STCG framework



STCG module



Dual 3D swin transformer blocks

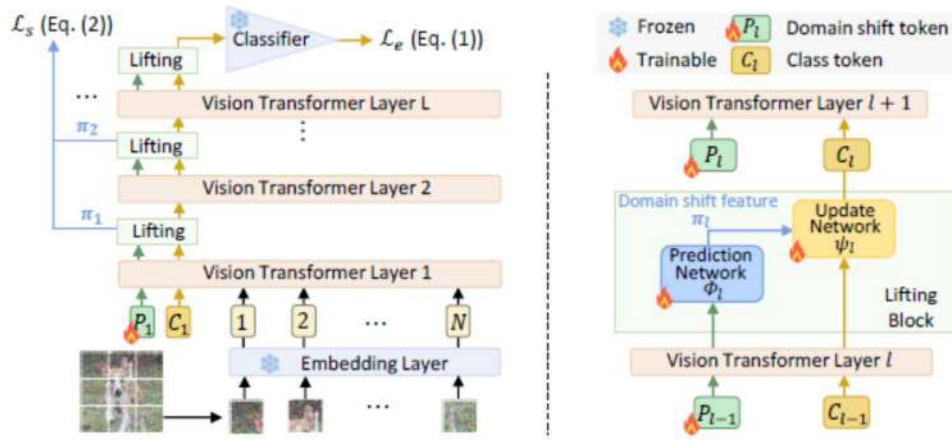


Fig. 2: An overview of the proposed Dual-Path Adversarial Lifting method. During inference in the target domain, the domain shift token P_l, the prediction network Φ_l, and the update network Ψ_l are updated given each mini-batch testing samples. The dual-path lifting transformer (**Left**). The details of the lifting block in each layer (**Right**).

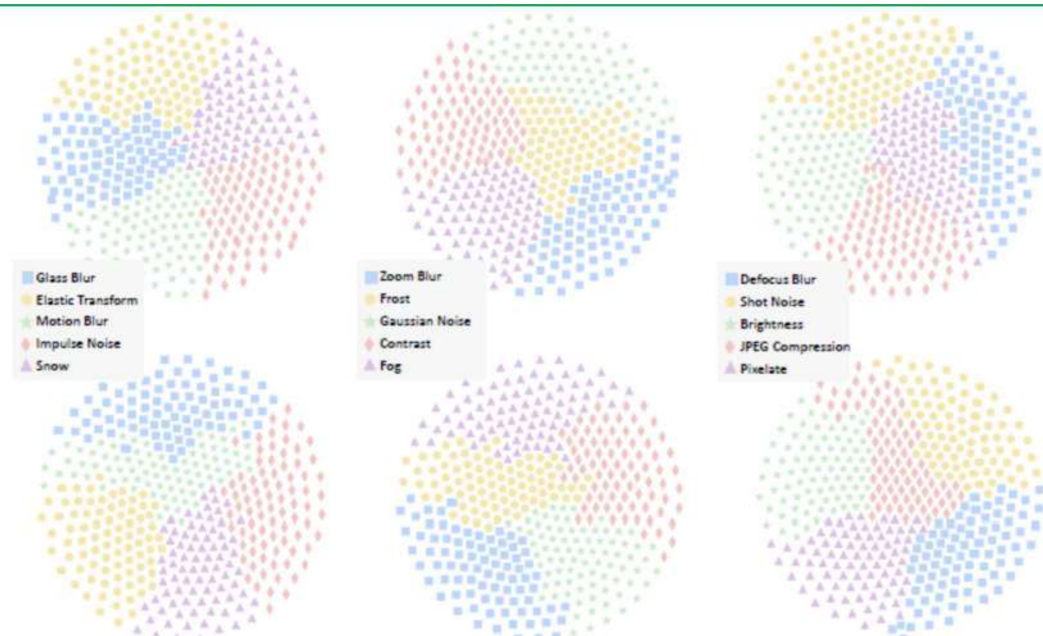
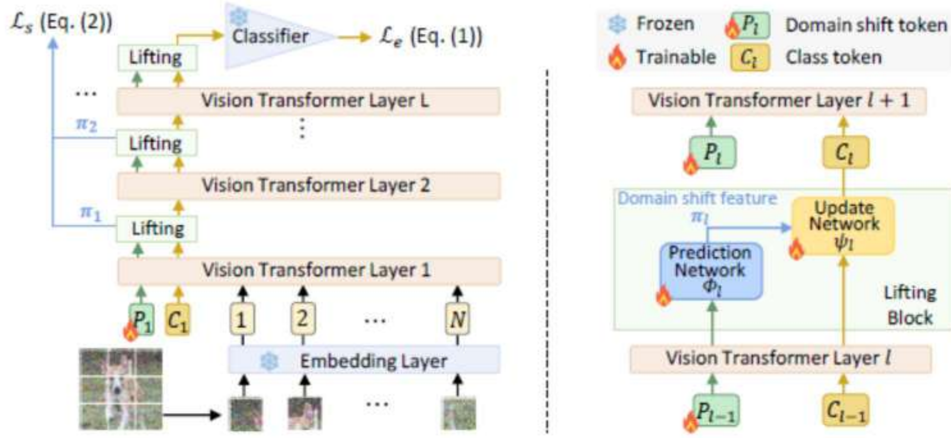
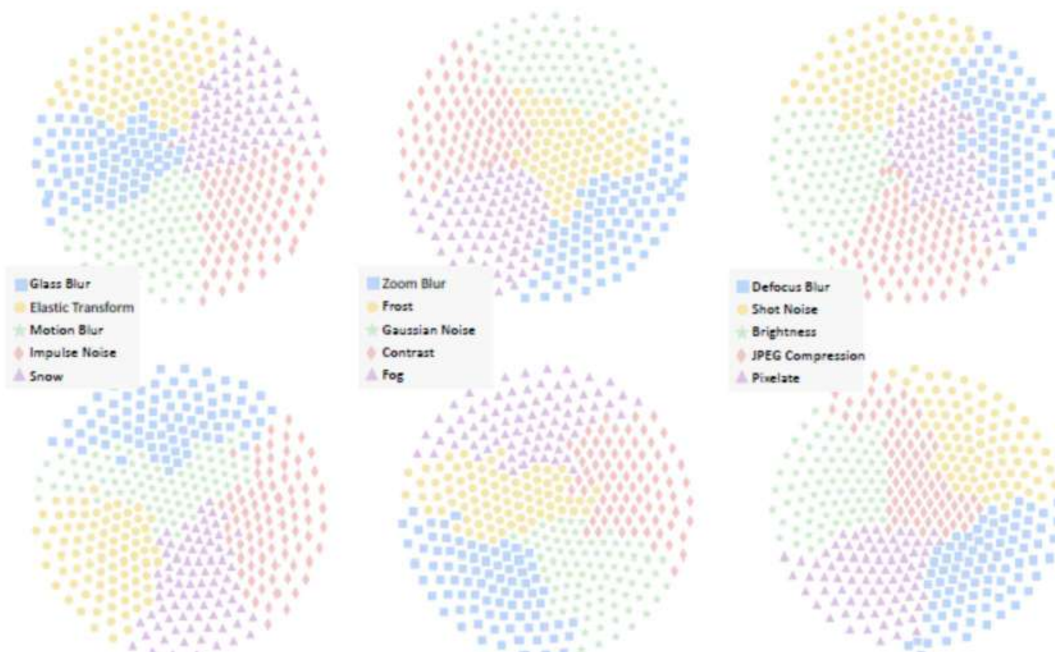


Fig. 4: The t-SNE visualization for the domain shift features in the first layer (top) and last layer (bottom) of ViT. Different colors represent different domains. It is able to learn domain-specific knowledge for different domains across layers.



An overview of the proposed Dual-Path Adversarial Lifting method.

✓ During inference in the target domain, the domain shift token Pl, the prediction network Φl, and the update network Ψl are updated given each mini-batch testing samples.
The dual-path lifting transformer (Left). The details of the lifting block in each layer (Right).



✓ **The t-SNE visualization for the domain shift features in the first layer (top) and last layer (bottom) of ViT. Different colors represent different domains. It is able to learn domain-specific knowledge for different domains across layers.**

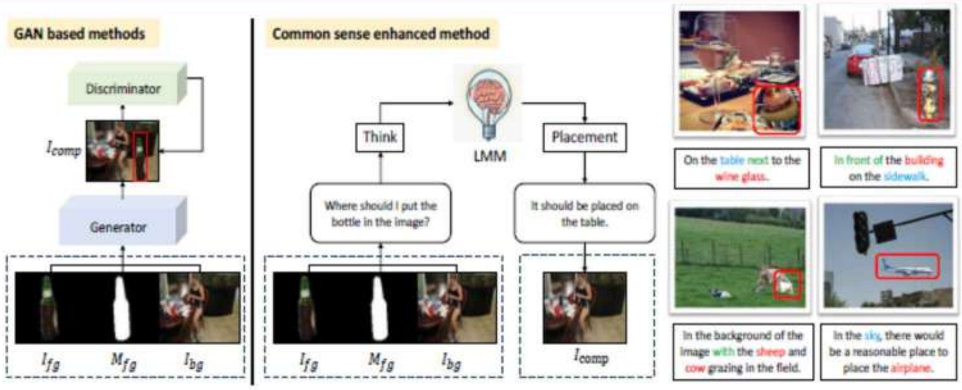
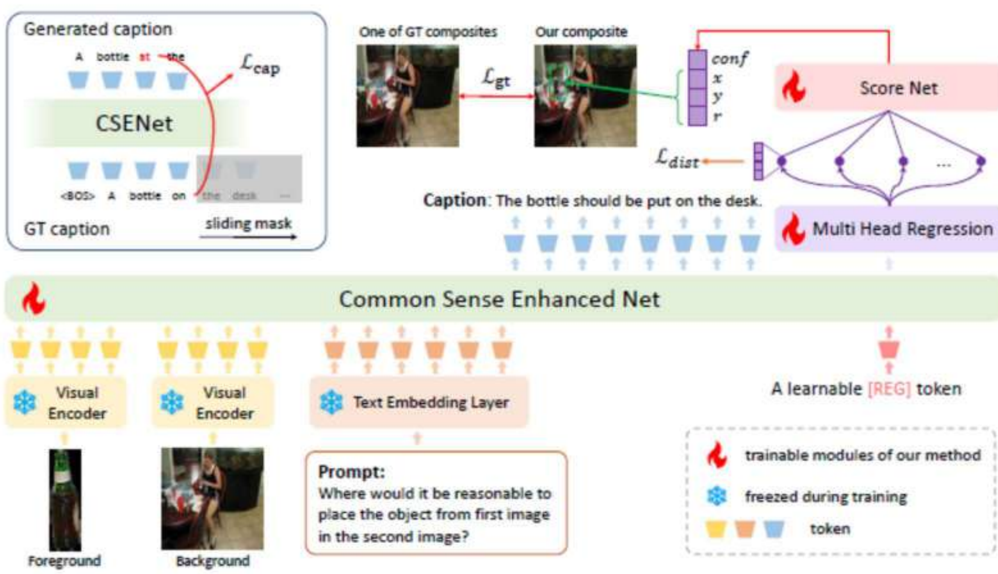
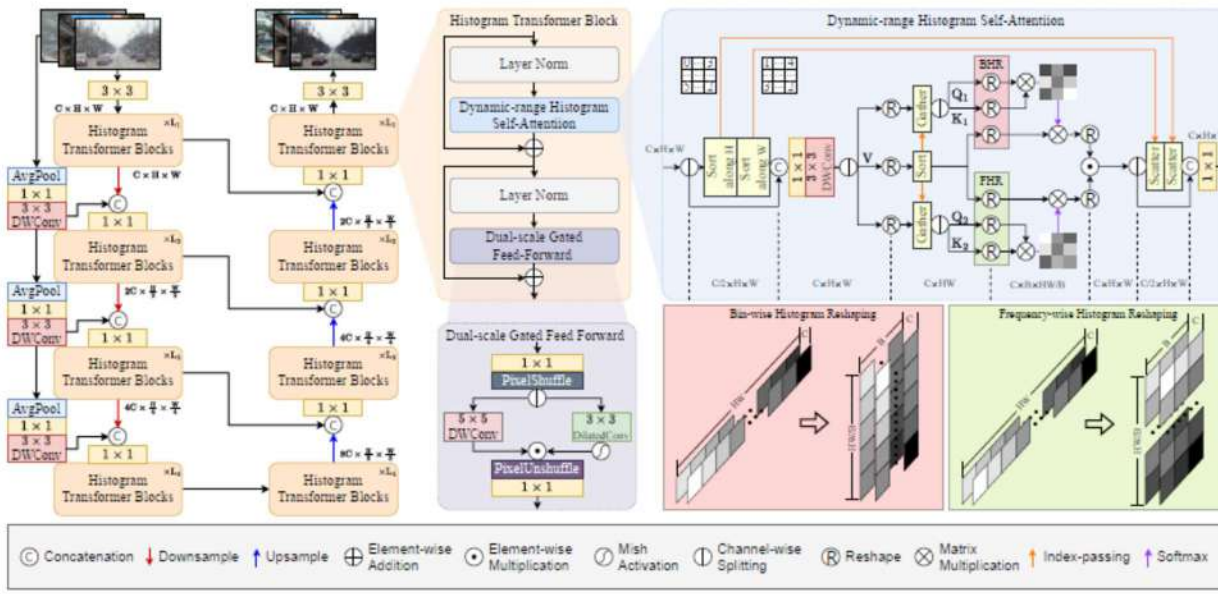


Fig. 1: Illustration of the *Think Before Placement* process. While other methods directly determine object placement based on image features, our approach adds a thinking process to this procedure. First, a large multi-modal model assesses which positions are reasonable, then it proceeds with the placement, achieving a more rational positioning of objects. I_{fg}, M_{fg}, I_{bg} and I_{comp} stands for foreground image, foreground mask, background image and composite image respectively.





The overall architecture of Histoformer for weather removal

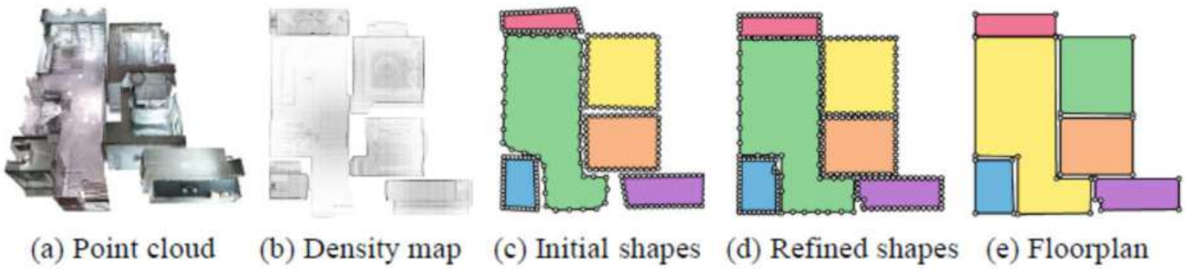
✓ The main component is the Histogram Transformer block, and it comprises the Dynamic-range Histogram Self-Attention (DHSA) module and the Dual-scale Gated Feed-Forward (DGFF) module. Within DHSA, we present two types of reshaping mechanism, i.e., Bin-wise Histogram Reshaping and Frequency-wise Histogram Reshaping

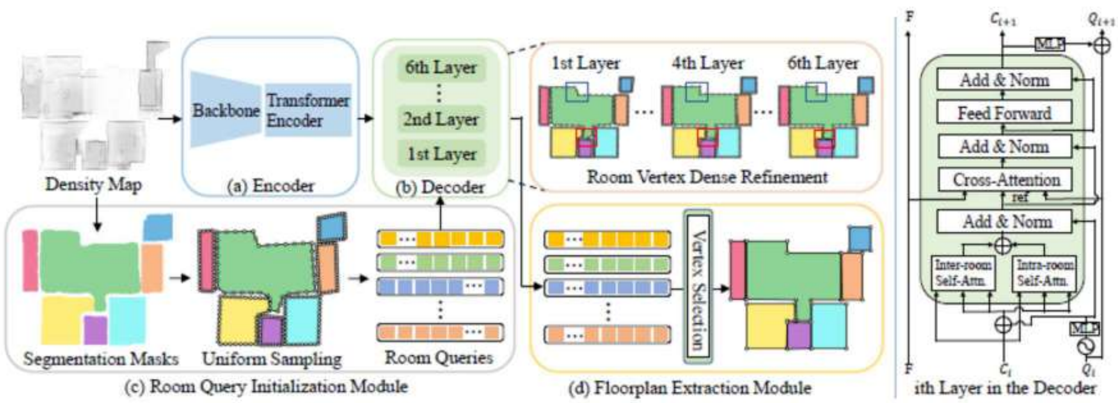
(a) Input (b) [86] (c) [70] (d) [100] (e) [53] (f) Ours (g) Clean

Fig. 3: Visual comparison for desnowing on Snow100K [44]. The samples from (b) to (e) are Restormer [86], TransWeather [70], WGWSNet [100], WeatherDiff [53].

(a) Input (b) [86] (c) [70] (d) [100] (e) [53] (f) Ours (g) Clean

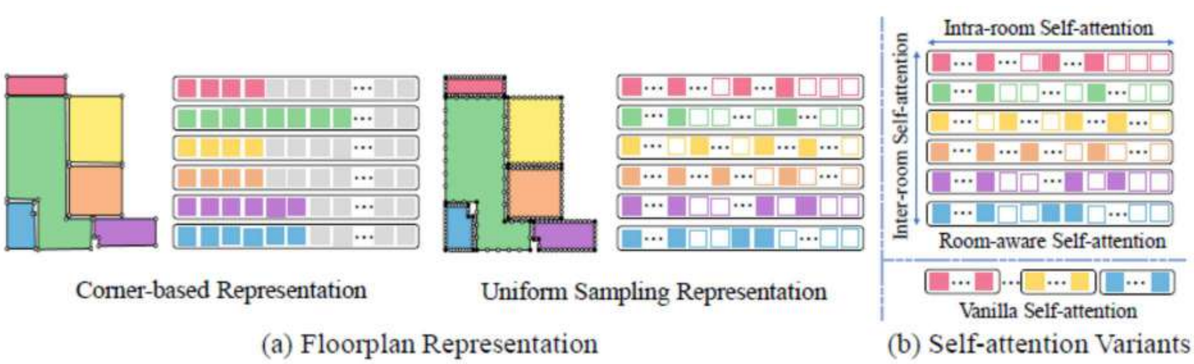
Fig. 4: Visual comparison for deraining and dehazing on Outdoor-Rain [32]. The samples from (b) to (e) are Restormer [86], TransWeather [70], WGWSNet [100], WeatherDiff [53].





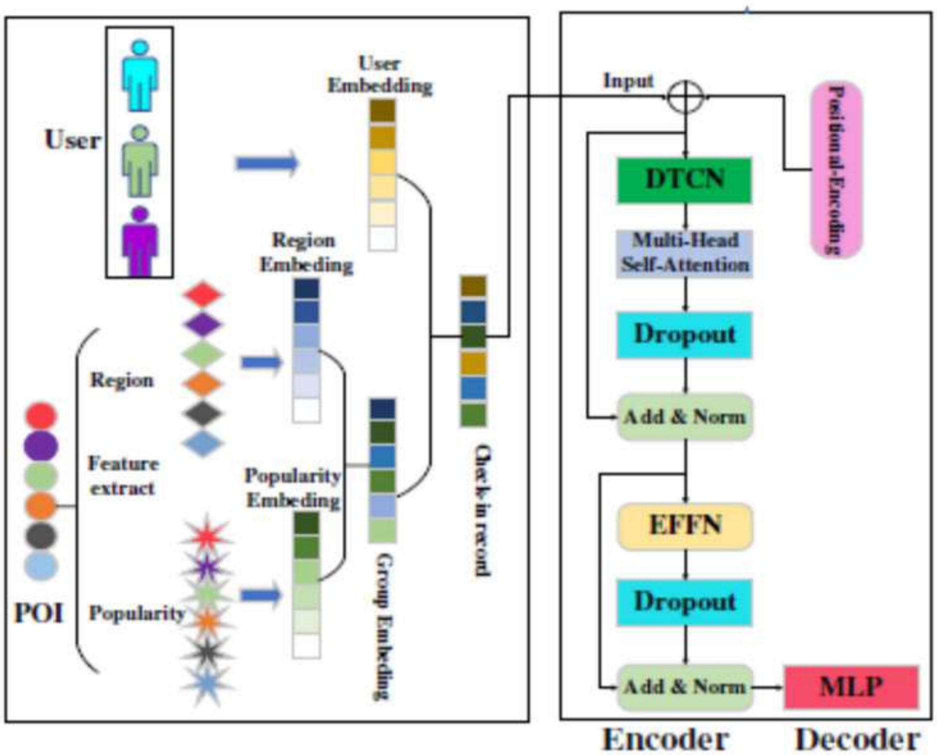
Overall architecture of PolyRoom.

✓ PolyRoom consists of four main components:
  - Encoder module,
  - Decoder module,
  - Room-aware query initialization module, and
  - Floorplan extraction module.
- Room queries are initialized with instance segmentation. Subsequently, they are refined in the Transformer decoder layer by layer with dense supervision (red and blue boxes mark the changes). Finally, floorplan is extracted based on vertex selection.
- The detailed structure of the ith layer in the Transformer decoder is depicted in the right part, where F denotes
- output of the Transformer encoder, Ci, Ci+1 represent content queries from different layers, while Qi, Qi+1 denote room queries from different layers

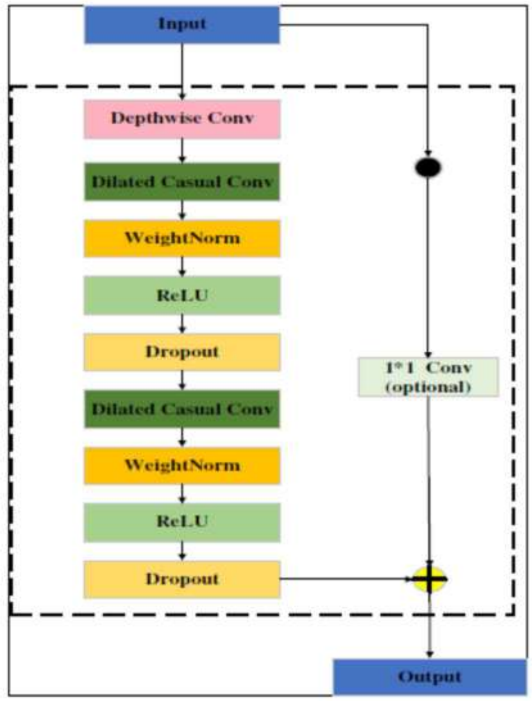


- (a) Illustration of floorplan representation including the sparse corner-based (left) and our dense uniform sampling representation (right). Valid contour vertices (outlined) and corner vertices (filled) with supervision during training are colored according to the room on the left.
- (b) Illustration of the self-attention variants including the room-aware self-attention and vanilla self-attention. Our room-aware self-attention is a combination of intra-room and inter-room self-attention, which works among different vertices in a single room and among different rooms. And the vanilla self-attention performs on the flattened queries.

Transformer Net	**2025-33**	



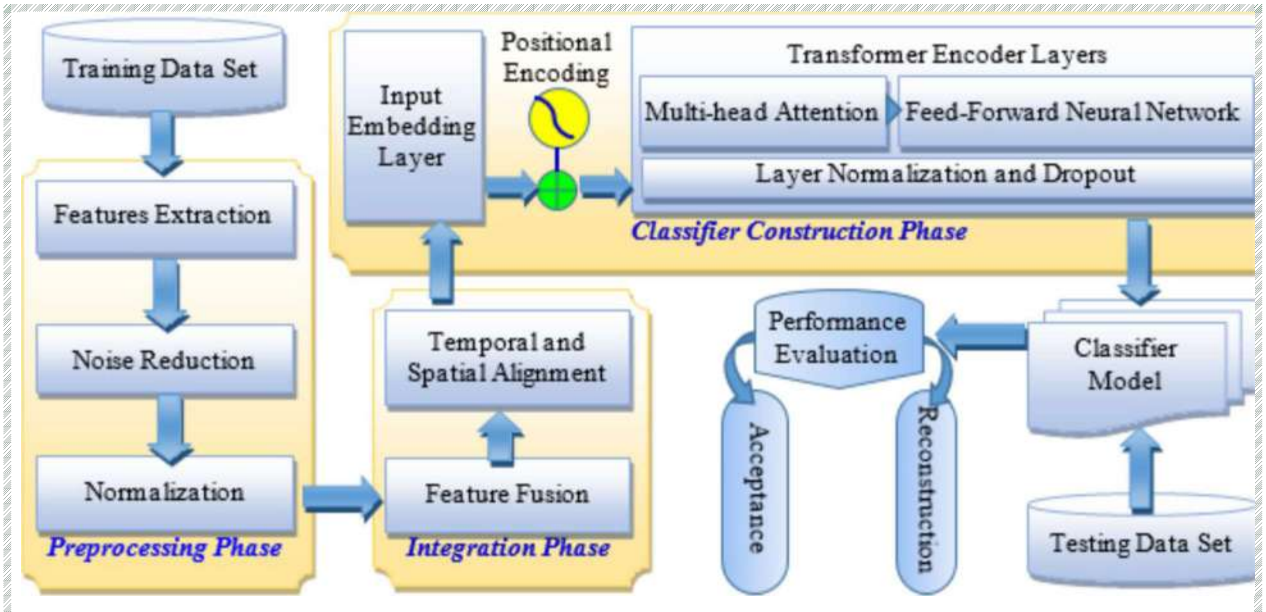
Overall structure diagram of DTCN-EFFN-Transformer model



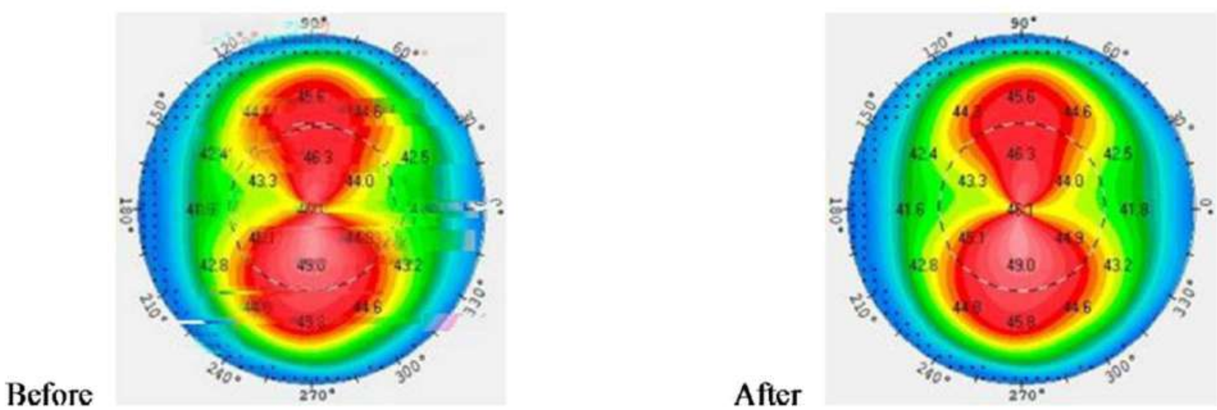
Deep time convolution network structure diagram

Structure diagram of EFFN

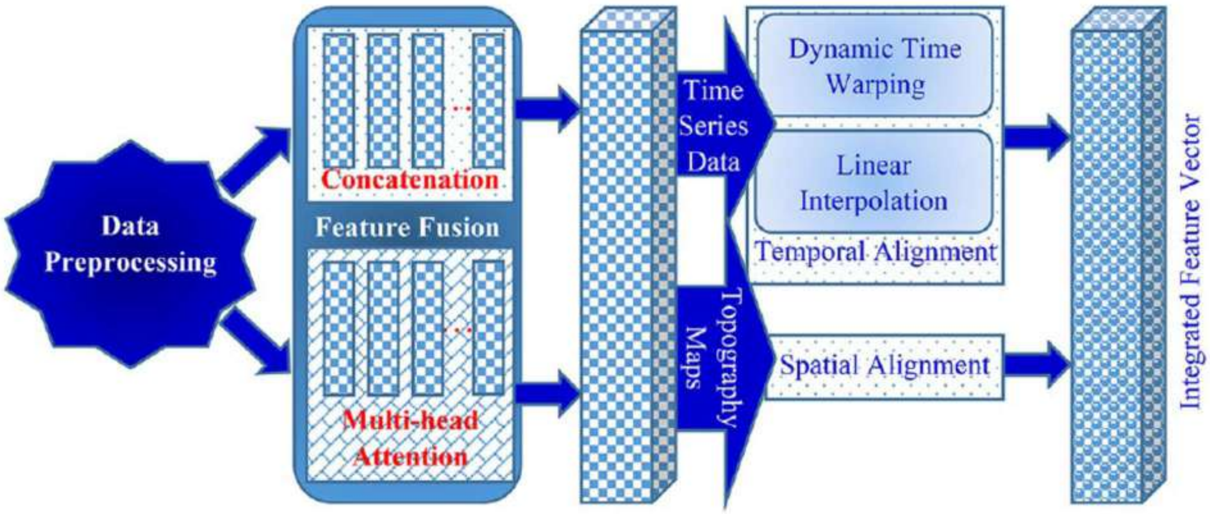
Transformer Net	**2025-35**	



The transformers-based multi-source keratoconus detection model architecture

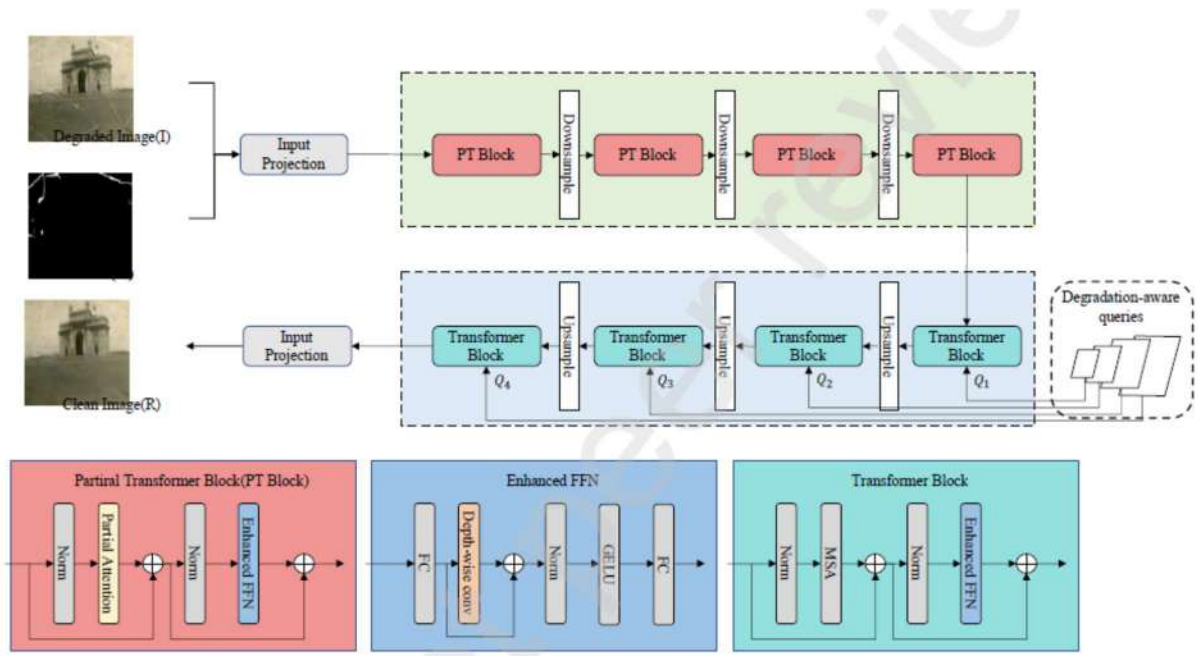


- ✓ **Corneal topography map before noise reduction in the left image and**
- ✓ **Corneal topography map after applying Gaussian filtering for noise reduction in the right image**



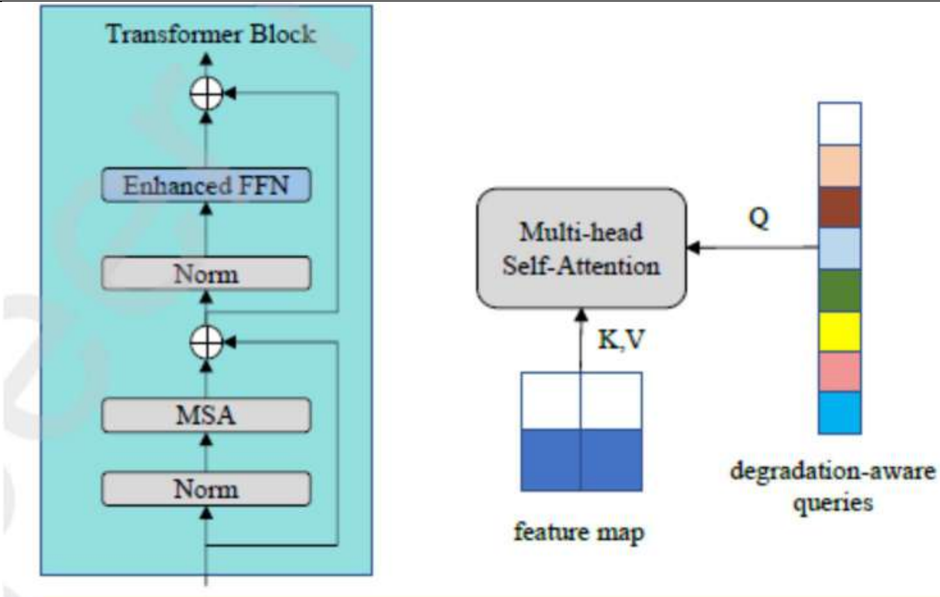
The data integration process

Transformer Net	**2025-36**	

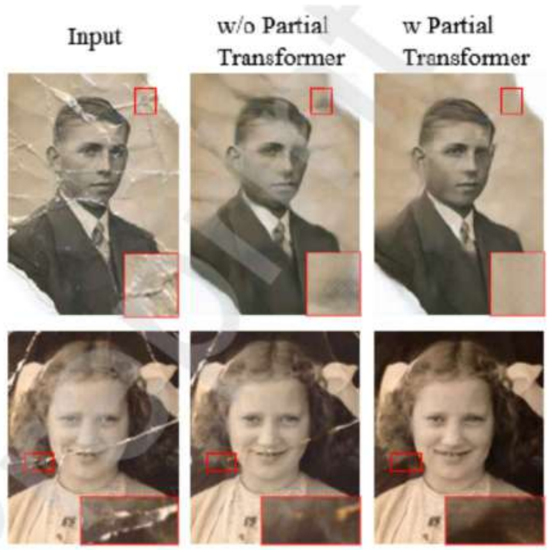


Overview of proposed MDTNet model.

- ✓ A degraded image is forwarded to the transformer encoder to extract hierarchical features.
- ✓ The encoder has Partial transformer blocks to extract useful features from the main patch.
- ✓ The transformer decoder has learnable degeneration-aware queries to obtain the task features.
- ✓ Then, hierarchical features from the encoder as well as the features from decoder are forwarded to a convolutional projection block to obtain the clean image



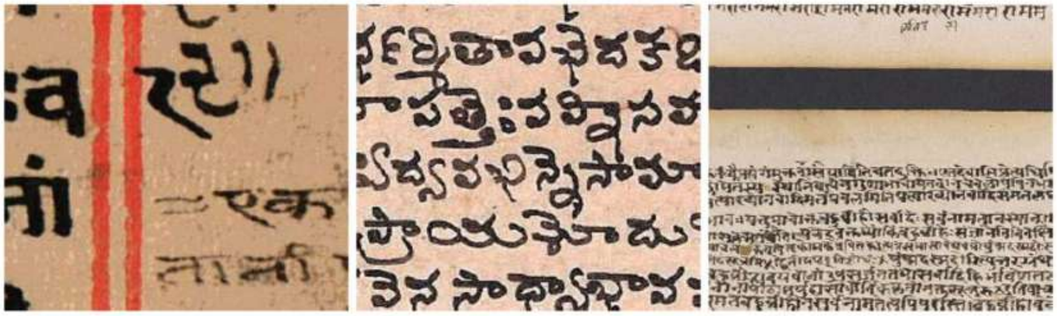
Configuration of the degradation-aware module and transformer decoder



Ablation study for the partial transformer block.

- ✓ **The results with the partial transformer block have fewer artifacts.**

Transformer Net	2025-37	



(a) too *little* context! (b) *good* context (c) too *much* context!

Fixed size patches (256 × 256) sampled from different documents can result in *poor context*

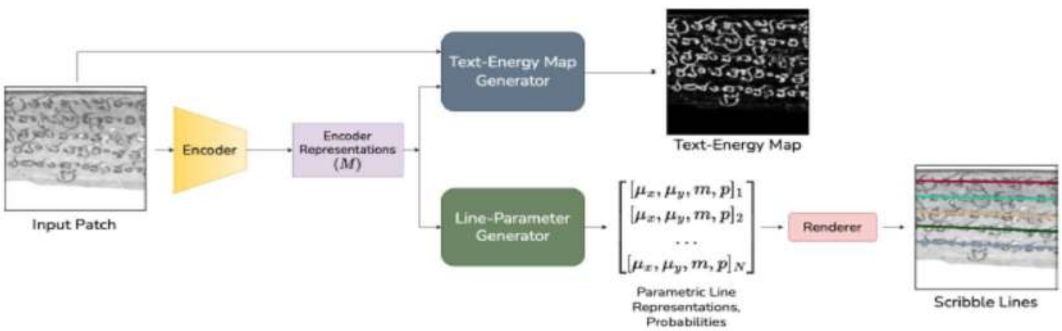
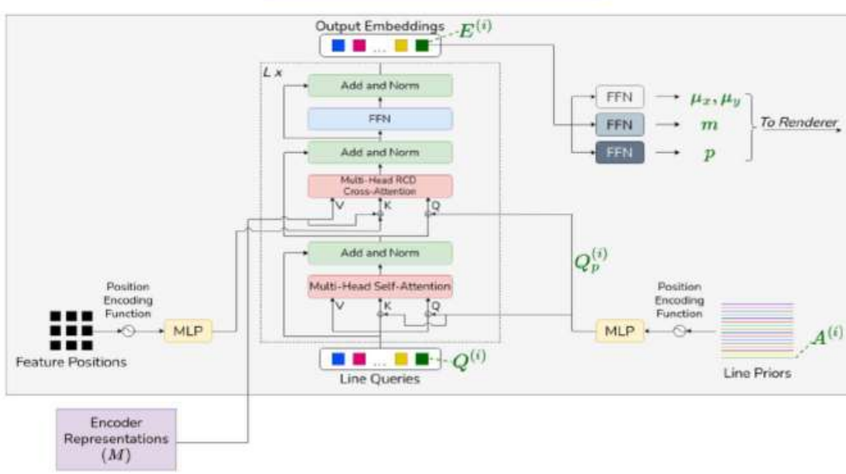


Figure 3: Stage-1 pipeline (Sec. 3.1). Scribbles lines are overlaid on the input patch for clarity.

Line-Parameter Generator



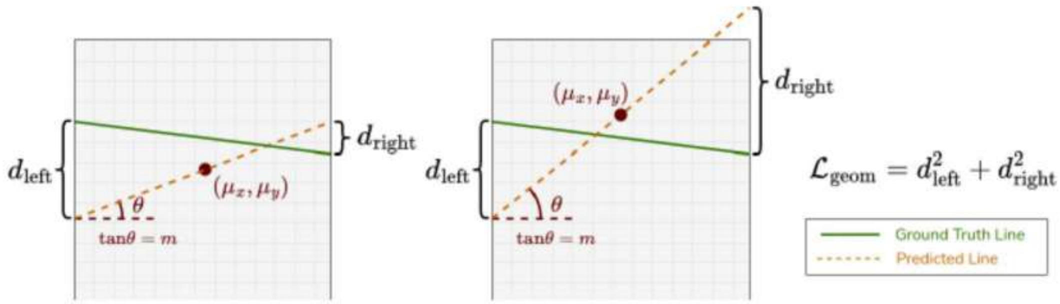


Figure 5: Our proposed loss function for penalizing the misalignment between two lines. If the scribble line touches the top or the bottom patch boundary, then we use interpolation to calculate the loss as shown in the figure on the right.

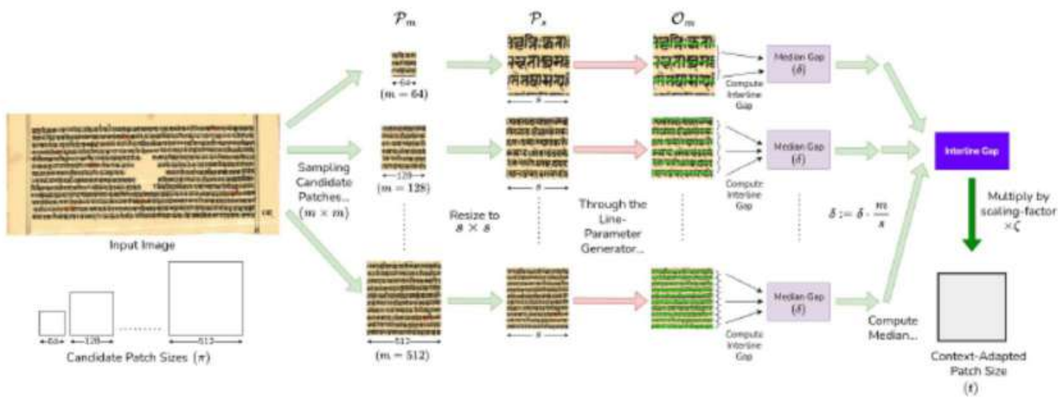
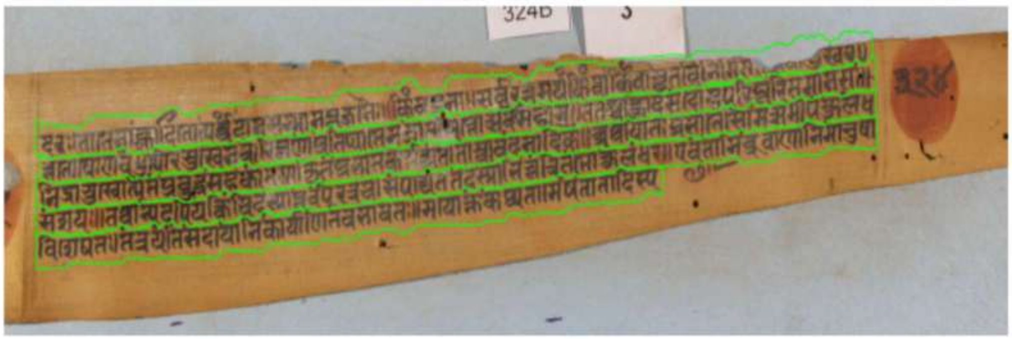


Figure 7: First, *candidate* patches of various sizes $m = \{64, 128, \ldots 512\}$ are sampled from the document. Line-parameter predictions are made for each patch by resizing them to the input shape expected by the model ($s = 256$) and then passing them through the Line-Parameter Generator (Sec. 3.2). An interline-gap value δ for each patch is calculated using the line predictions for the patch. These patch-level interline gap predictions are then scaled to the size of the original document ($\delta := \delta \cdot \frac{m}{s}$) and averaged to obtain an estimate of the average interline gap value for the whole document. This averaged value is multipled by a scaling factor ζ, which roughly represents the expected number of text lines seen in a patch to obtain the final context-adapted patch size (t). Patches of size t are then sampled from the original document, and are fed to the Line-Parameter Generator to get the final predictions.

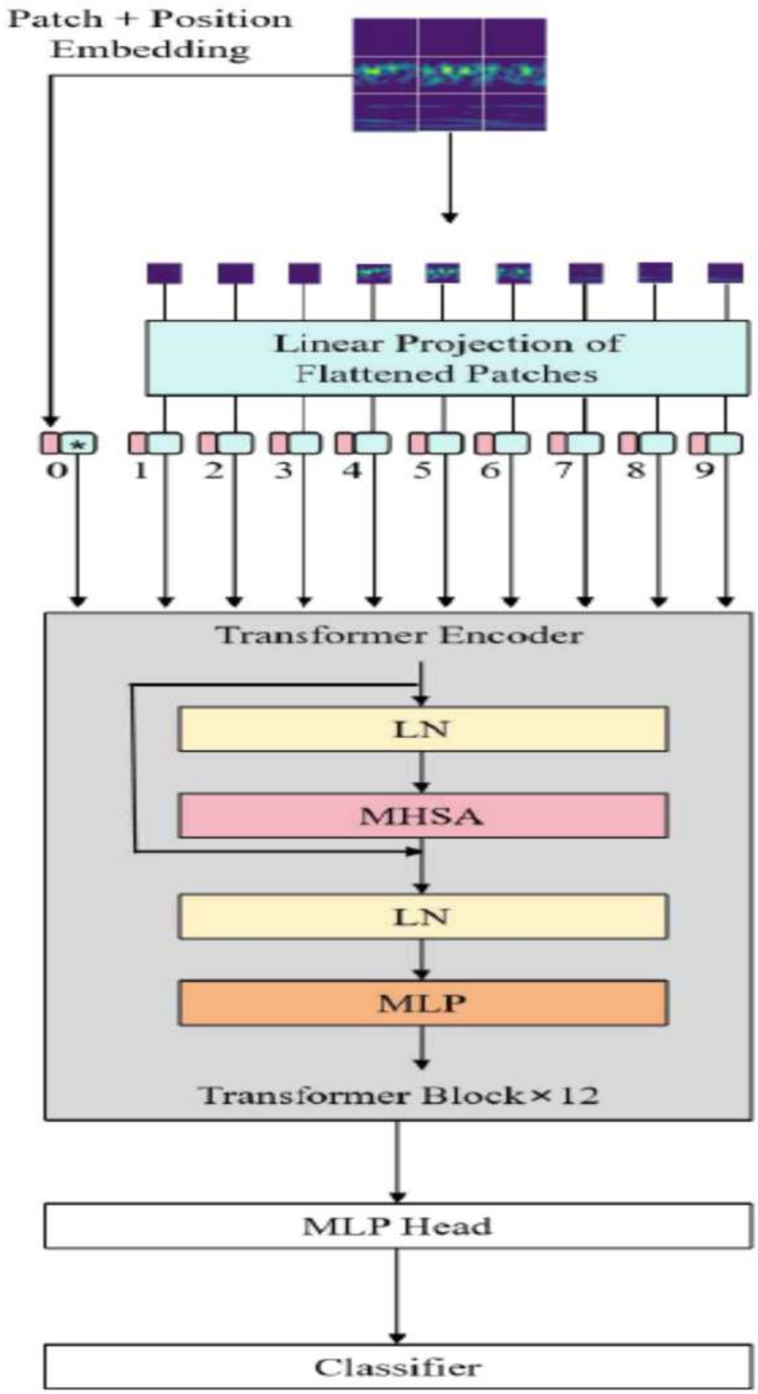
(a) WM



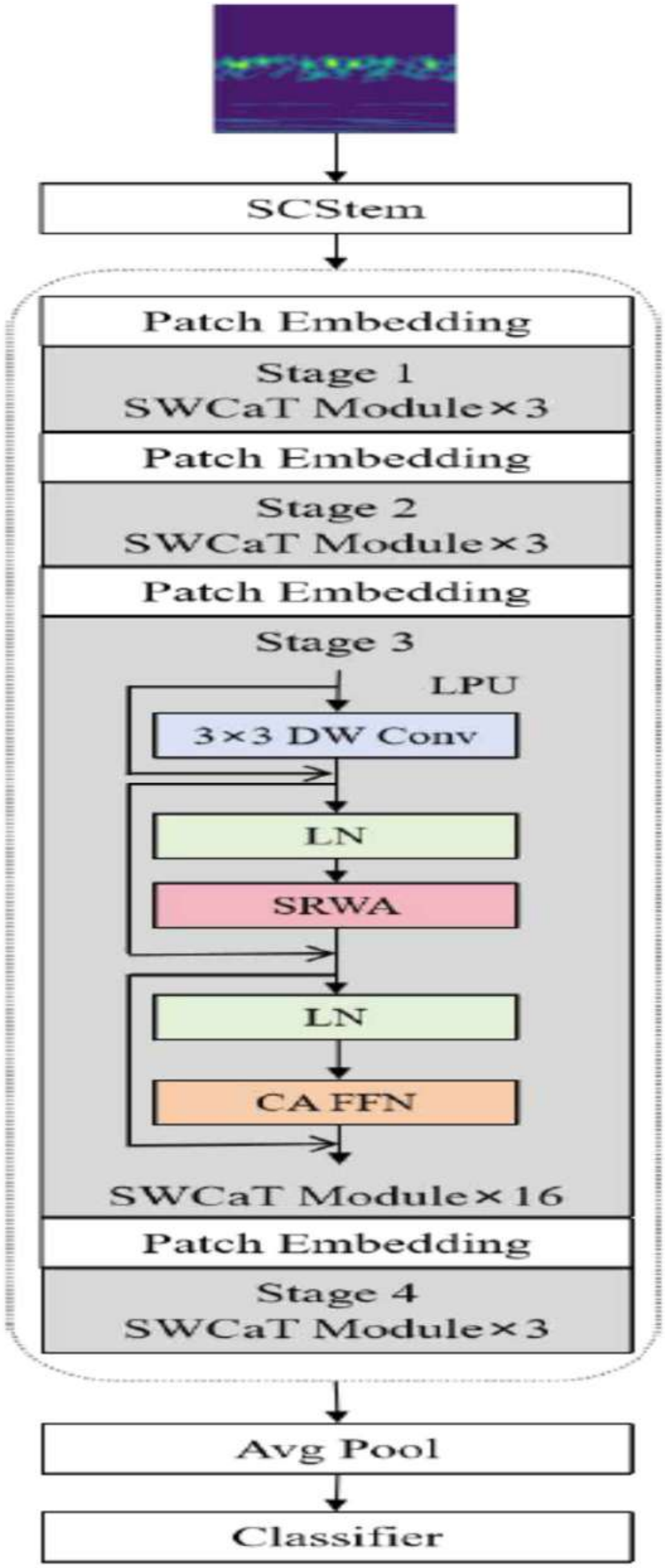
(b) UB

(c) SM

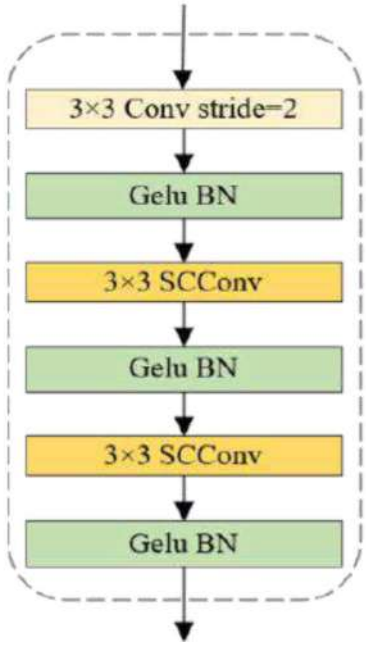
Zero-shot outputs of LineTR on the newly introduced datasets



Structure of the vision transformer (ViT-S).



Structure of novel vision transformer (EHVT)



Structure of SCStem module

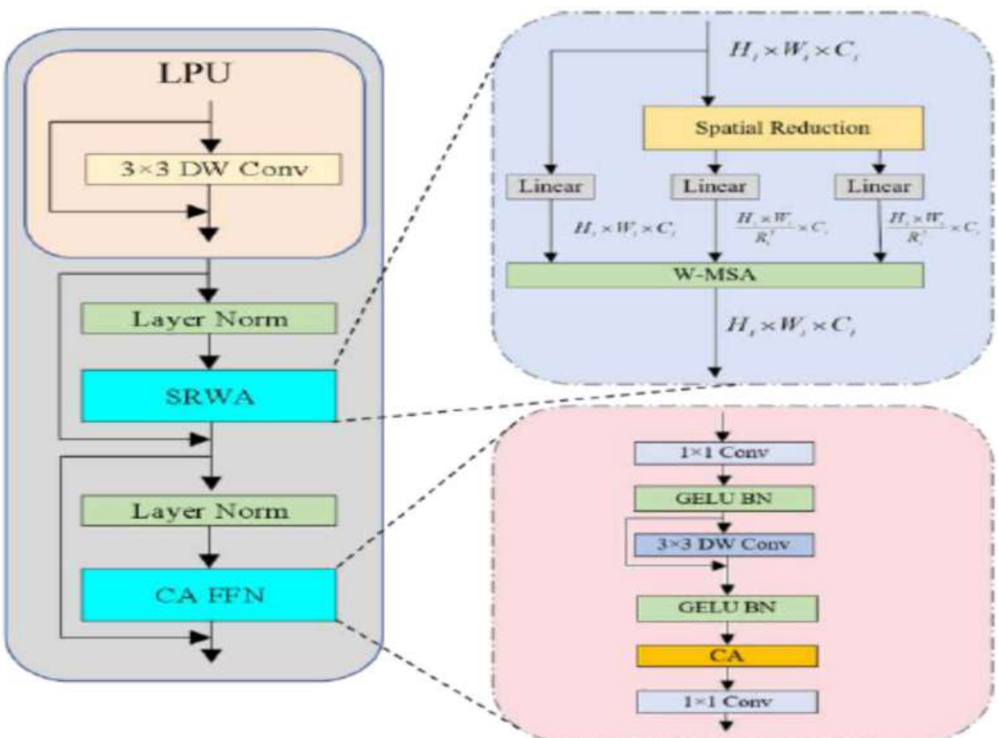


Figure 5. Structure of the SWCaT Module.

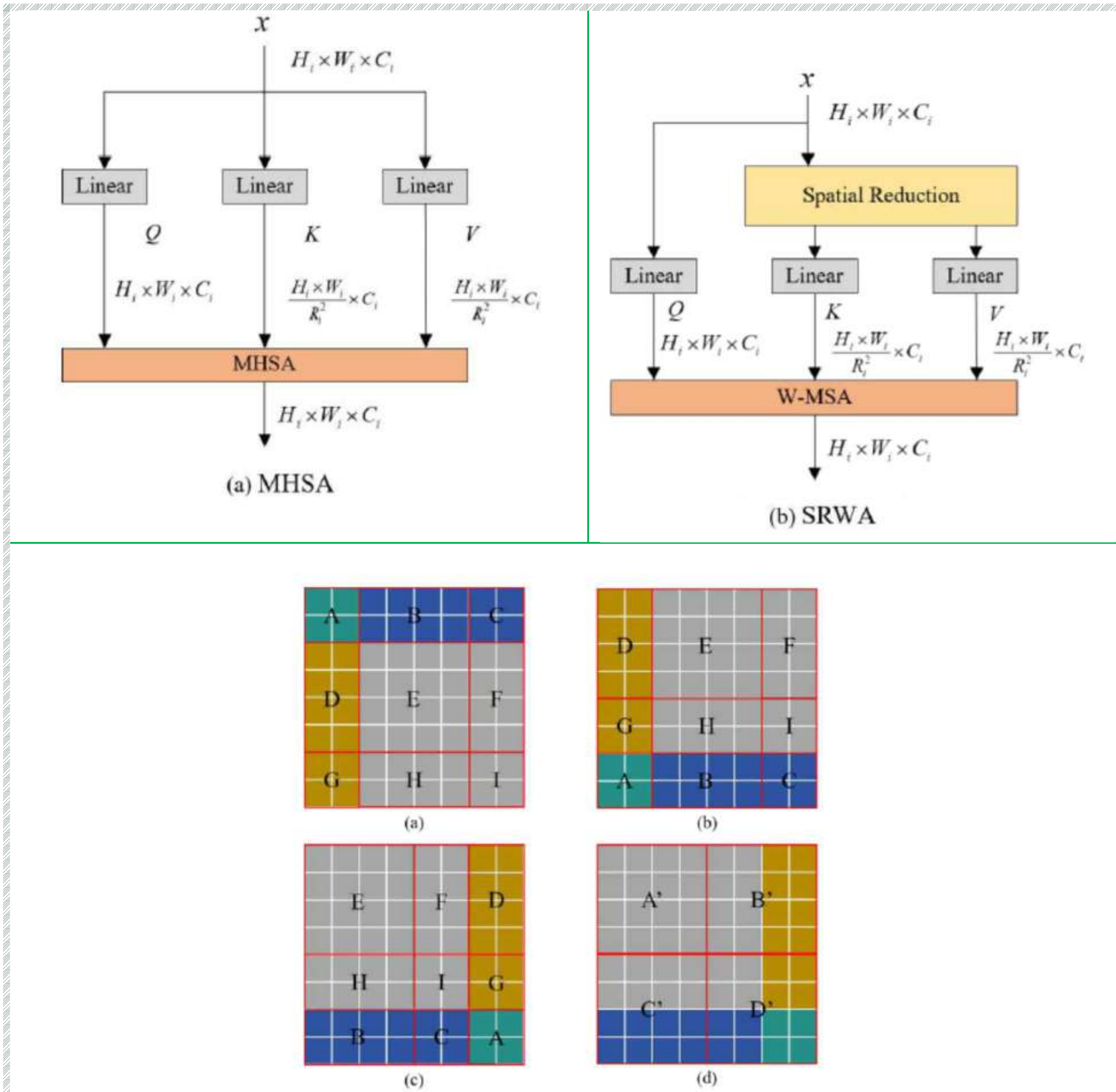


Figure 7. Schematic of SRWA module divide and shift windows

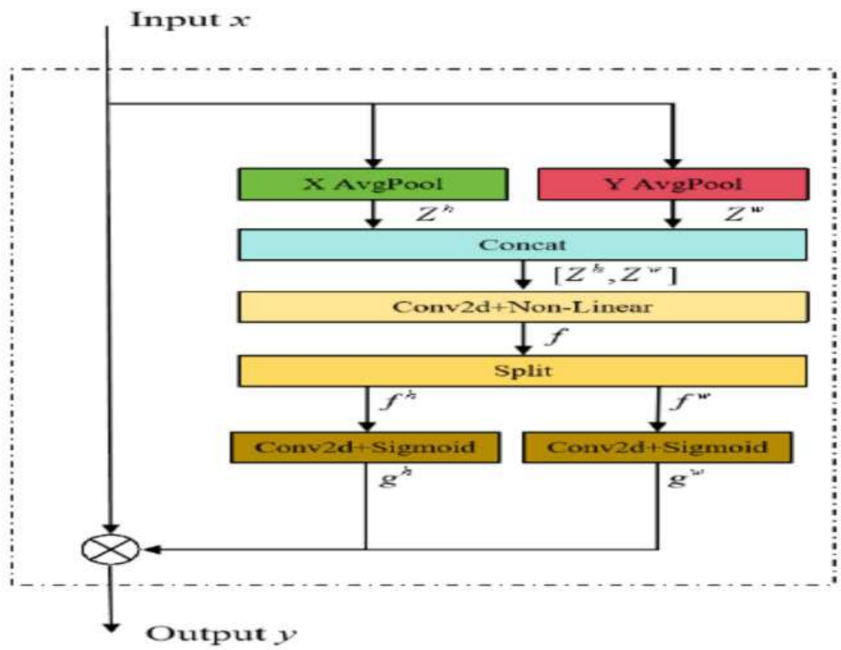


Figure 8. Structure of the CA block.

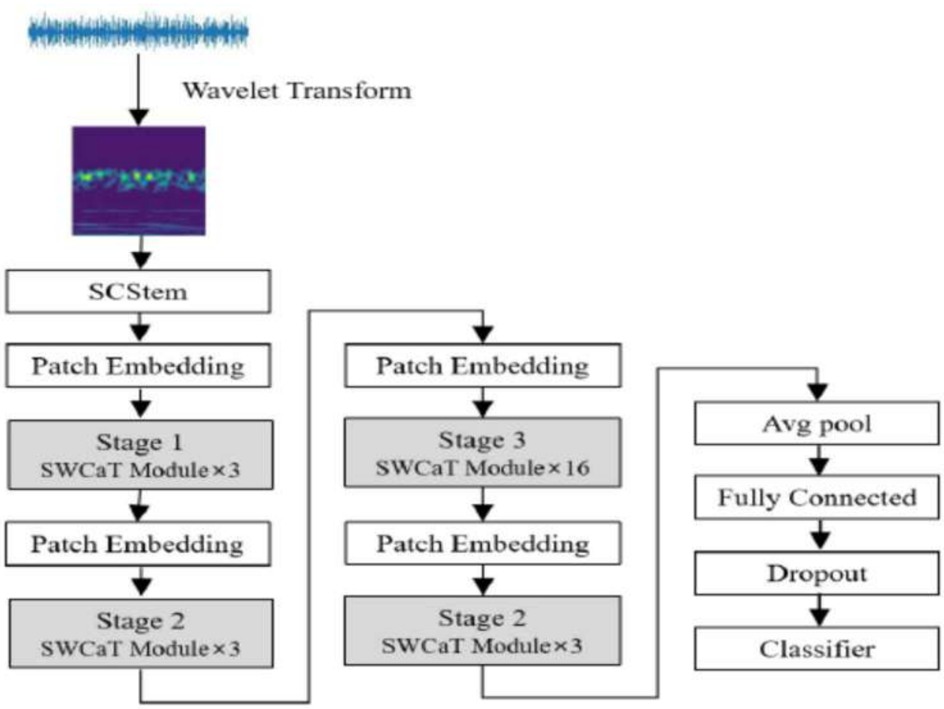
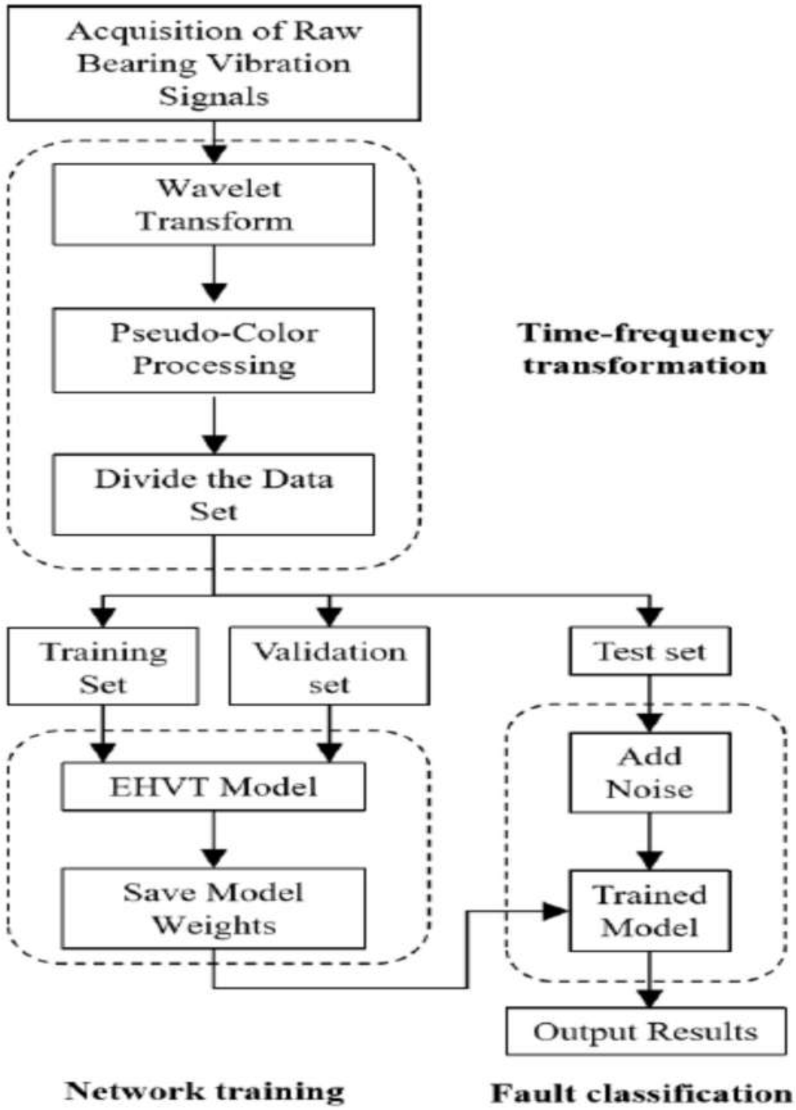
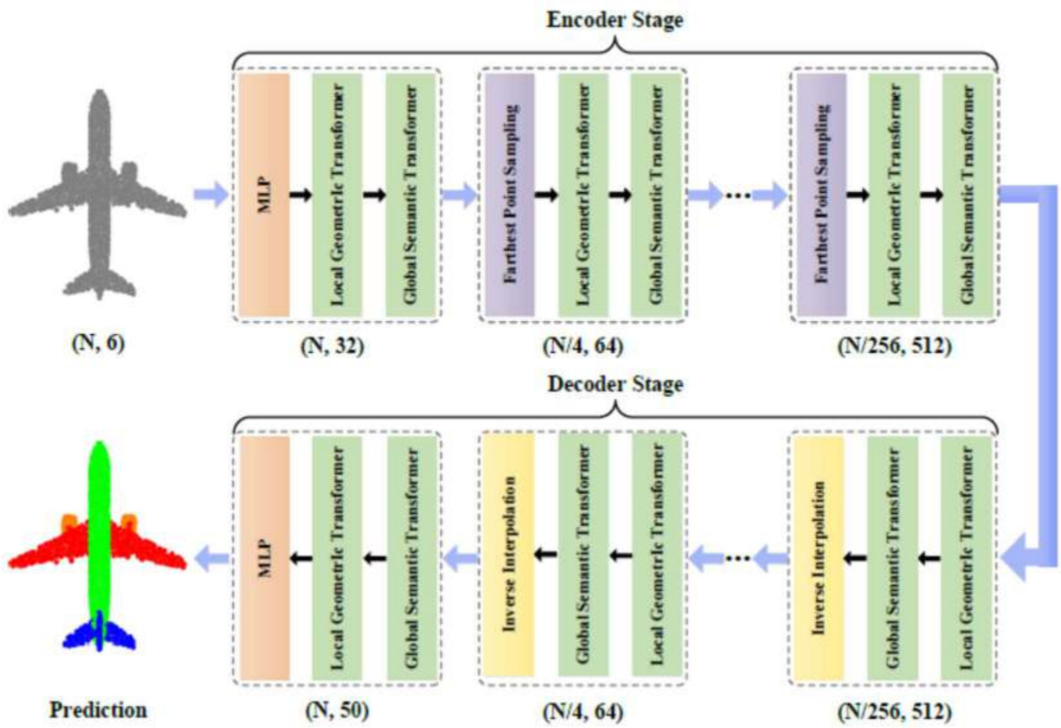


Figure 9. Frame of the proposed enhanced model.

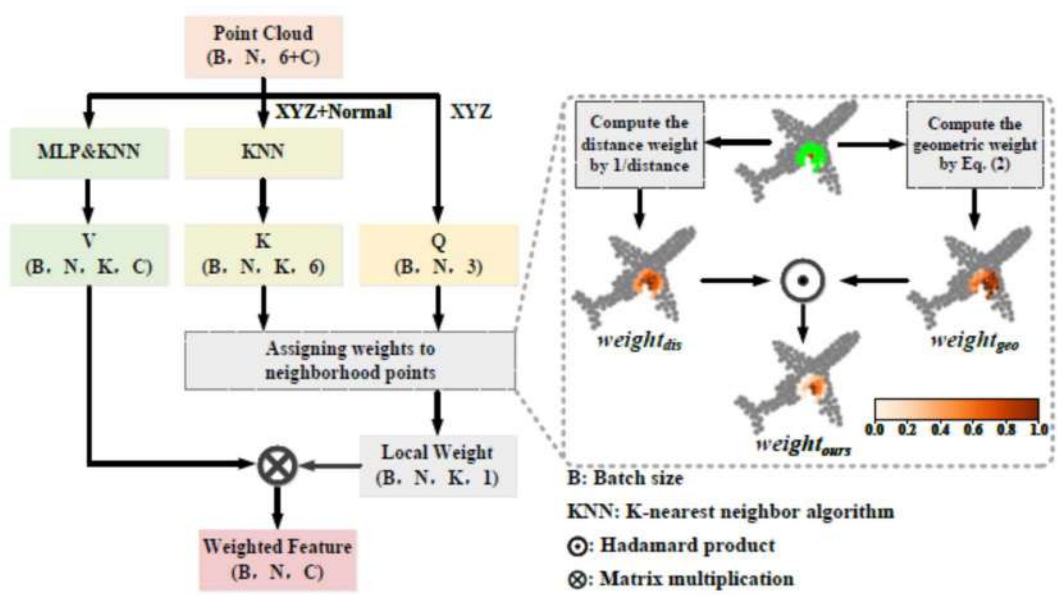


Flowchart of fault diagnosis



Overview of the proposed model.

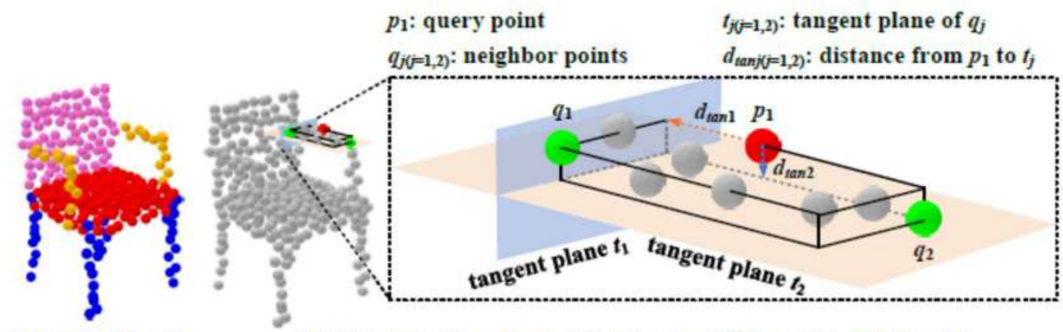
- ✓ In the encoder and decoder stages, the transformer structure serves as the primary feature aggregator throughout the network
- ✓ MLP: Multi-layer perception. N: The number of points in the point cloud



Structure of the local geometric transformer module.

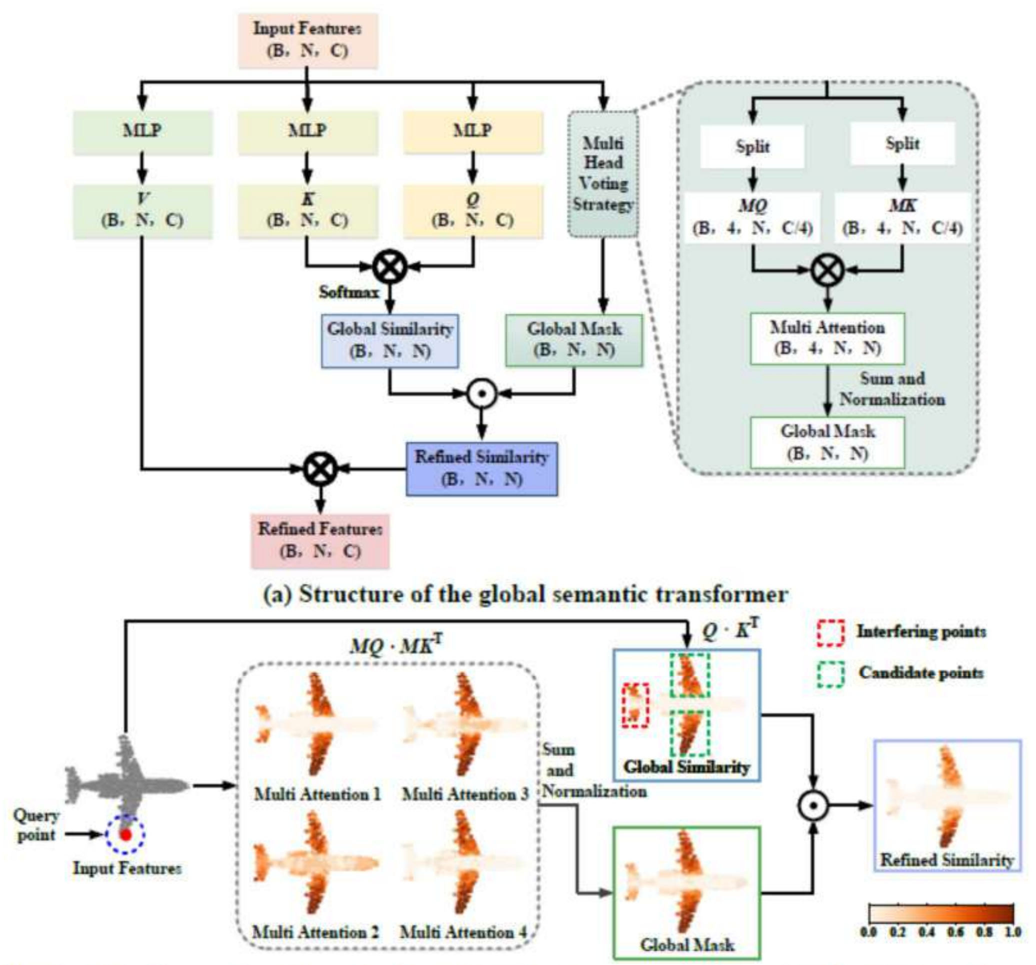
- ✓ We visualize the local weight on an airplane, with a red query point located on the wing.

- ✓ In weightours, the weights of neighbor points in the wing decay slowly as the distance increases. However, the weights of other neighbor points decay rapidly



(a) Ground truth (b) Magnification of tangent planes for different neighborhoods

- **Illustration of the distance dtan1 and dtan2 from p1 to the tangent plane of q1 and q2, respectively.**
- **Although the Euclidean distance from p1 to both q1 and q2 remain the same, p1 is closer to t2, signifying that q2 holds greater significance than q1.**



(a) Structure of the global semantic transformer

(b) Pipeline of generating the refined similarity by incorporating the multi-head voting strategy

Overview of the global semantic transformer module.

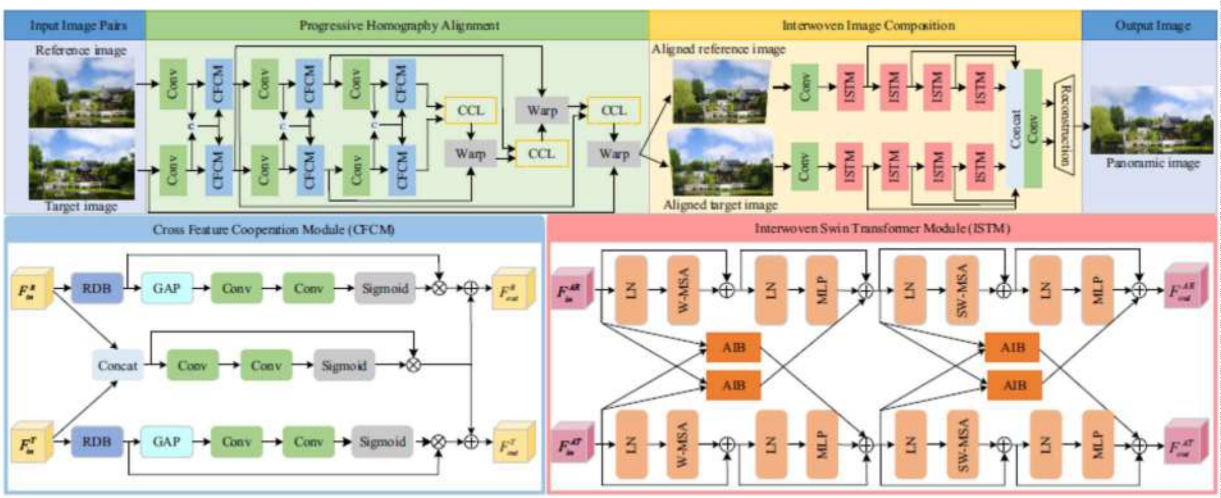
- ✓ **We visualize the refined**
- ✓ **similarity on an airplane, with a red query point located at the wing.**
- ✓ **In the refined similarity generated by our method, high response weights are exclusively assigned to points belonging to the wing section**

Table 2. Part segmentation results on ShapeNetPart dataset.

Methods	Ins. mIoU	Cat. mIoU	air.	bag	cap	car	cha.	ear.	gui.	kni.	lam.	lap.	mot.	mug	pis.	roc.	ska.	tab.
PointAttn. [5]	85.9	84.1	83.3	86.1	85.7	80.3	90.5	82.7	91.5	88.1	85.5	95.9	77.9	95.1	84.0	64.3	77.6	82.8
PointASNL[26]	86.1	83.4	84.1	84.7	87.9	79.7	92.2	73.7	91.0	87.2	84.2	95.8	74.4	95.2	81.0	63.0	76.3	83.2
PointMLP [12]	86.1	84.6	83.5	83.4	87.5	80.5	90.3	78.2	92.2	88.1	82.6	96.2	77.5	**95.8**	85.4	**64.6**	**83.3**	84.3
APES [25]	85.8	83.9	85.3	85.8	88.1	**81.2**	90.6	74.0	90.4	**88.7**	85.1	95.8	76.1	94.2	83.1	61.1	79.3	84.2
PointTran. [30]	86.5	83.7	85.8	85.3	86.8	77.2	90.5	82.0	90.8	87.5	85.2	96.3	75.4	93.5	83.8	59.7	77.5	82.5
PointGT [9]	85.8	84.2	84.3	84.5	88.3	80.9	91.4	78.1	92.1	88.5	85.3	95.9	77.1	95.1	84.7	63.3	75.6	81.4
PCT [7]	86.4	83.1	85.0	82.4	89.0	**81.2**	91.9	71.5	91.3	88.1	86.3	95.8	64.6	95.8	83.6	62.2	77.6	83.7
LGGCM [3]	86.7	84.8	85.1	85.9	**90.3**	80.8	91.6	75.4	**92.7**	88.1	86.5	96.1	77.0	94.2	84.5	63.6	80.2	84.3
Ours	**87.5**	**85.6**	**86.1**	**87.2**	88.1	79.4	**92.4**	**82.3**	92.0	88.4	**86.9**	**96.7**	**78.7**	95.6	**85.8**	63.8	79.3	**85.3**

Note: air.: airplane. cha.: chair. ear.: ear-phone. gui.: guitar. kni.: knife. lam.: lamp. lap.: laptop. mot.: motorbike. pis.: pistol. roc.: rocket. ska.: skateboard. tab.: table.

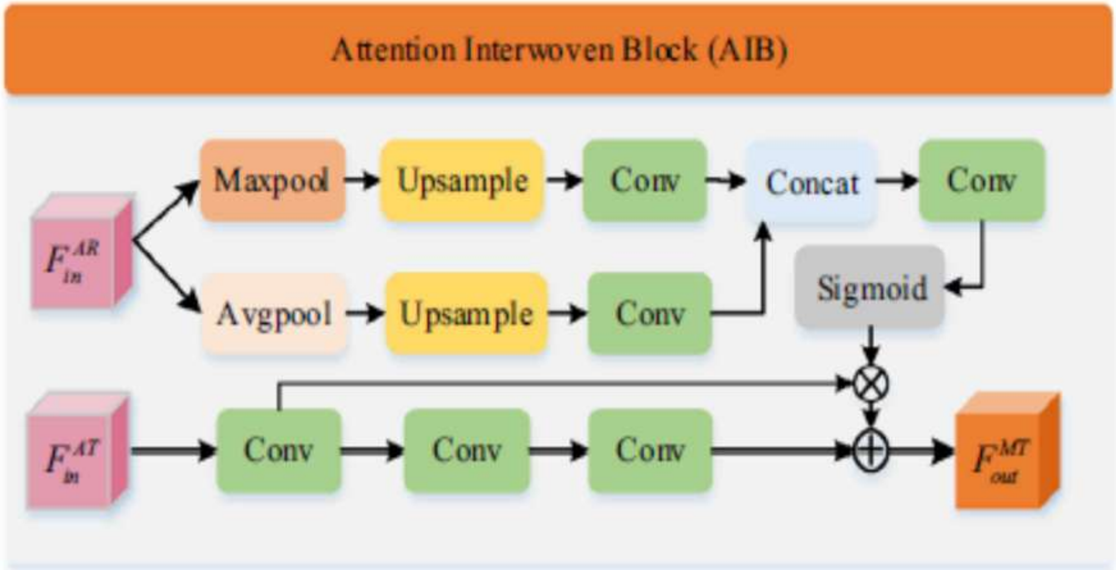
Transformer Net	**2025-41**	



Overview of the proposed PAIC-Net

PAIC-Net : Progressive Alignment and Interwoven Composition Network

Details of the attention interwoven block (AIB)



Transformer Net	2025-42	

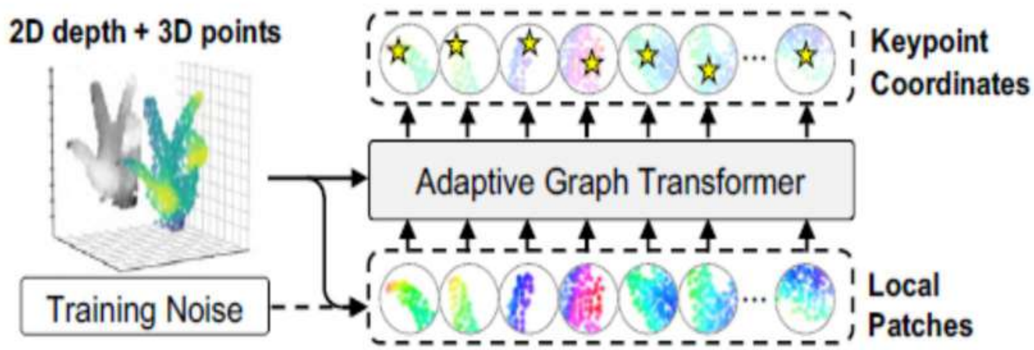
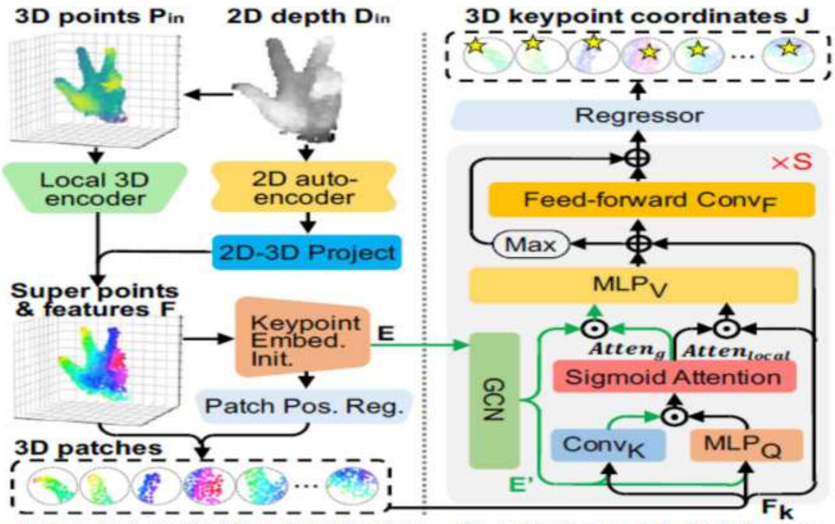


Illustration of the HandDAGT concept.

- DAGT: Denoising Adaptive Graph Transformer

- The model feeds 3D local patches cropped from input depth images and corresponding point clouds to the adaptive graph transformer for the keypoint coordinate estimation.
- The local patches are disturbed by the random Gaussian noise during training for a robust performance



(a) Patch & Embedding Initialization (b) Adaptive Graph Transformer

HandDAGT architecture.

- HandDAGT takes a 2D depth image and the sub-sampled point cloud as the input.
- The PointNet-based local 3D encoder and 2D auto-encoder extracts local 3D features and local 2D features, respectively.
- Then, the 2D features are projected into 3D space to fuse with 3D features forming the super point features F.
- Based on the super points, keypoint embeddings E and 3D patches are extracted as input to the novel adaptive graph transformer to estimate accurate 3D keypoint coordinates by leveraging the dynamic kinematic correspondences and local details.
- Notably, during the training stage, the local 3D patches are shifted by random noises in order to enforce the model providing robust estimations

Table 1: Comparison of the proposed method with previous state-of-the-art methods on the single-hand ICVL and NYU datasets. Input indicates the input type of 2D depth image (D), 3D voxels (V), or 3D point cloud (P).

Method	Mean keypoint error (mm)		Input
	ICVL	NYU	
Ren-9x6x6 [17]	7.31	12.69	D
DeepPrior++ [32]	8.1	12.24	D
Pose-Ren [4]	6.79	11.81	D
DenseReg [44]	7.3	10.2	D
CrossInfoNet [10]	6.73	10.08	D
JGR-P2O [11]	6.02	8.29	D
SSRN [37]	6.01	7.37	D
PHG [36]	5.97	7.39	D
HandPointNet [13]	6.94	10.54	P
Hand-Transformer [21]	6.47	9.80	P
Point-to-Point [16]	6.3	9.10	P
V2V [31]	6.28	8.42	V
HandFolding [8]	5.95	8.58	P
HandR2N2 [6]	5.70	7.27	P
IPNet [35]	5.76	7.17	D+P
HandDAGT (Ours)	**5.66**	**7.12**	D+P

20

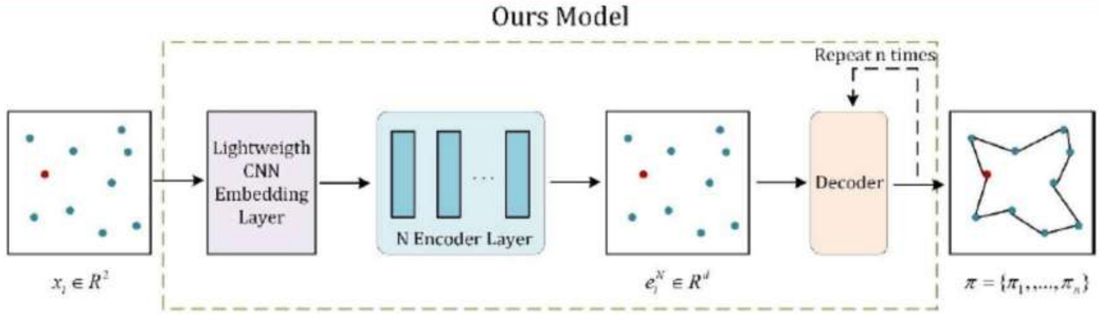
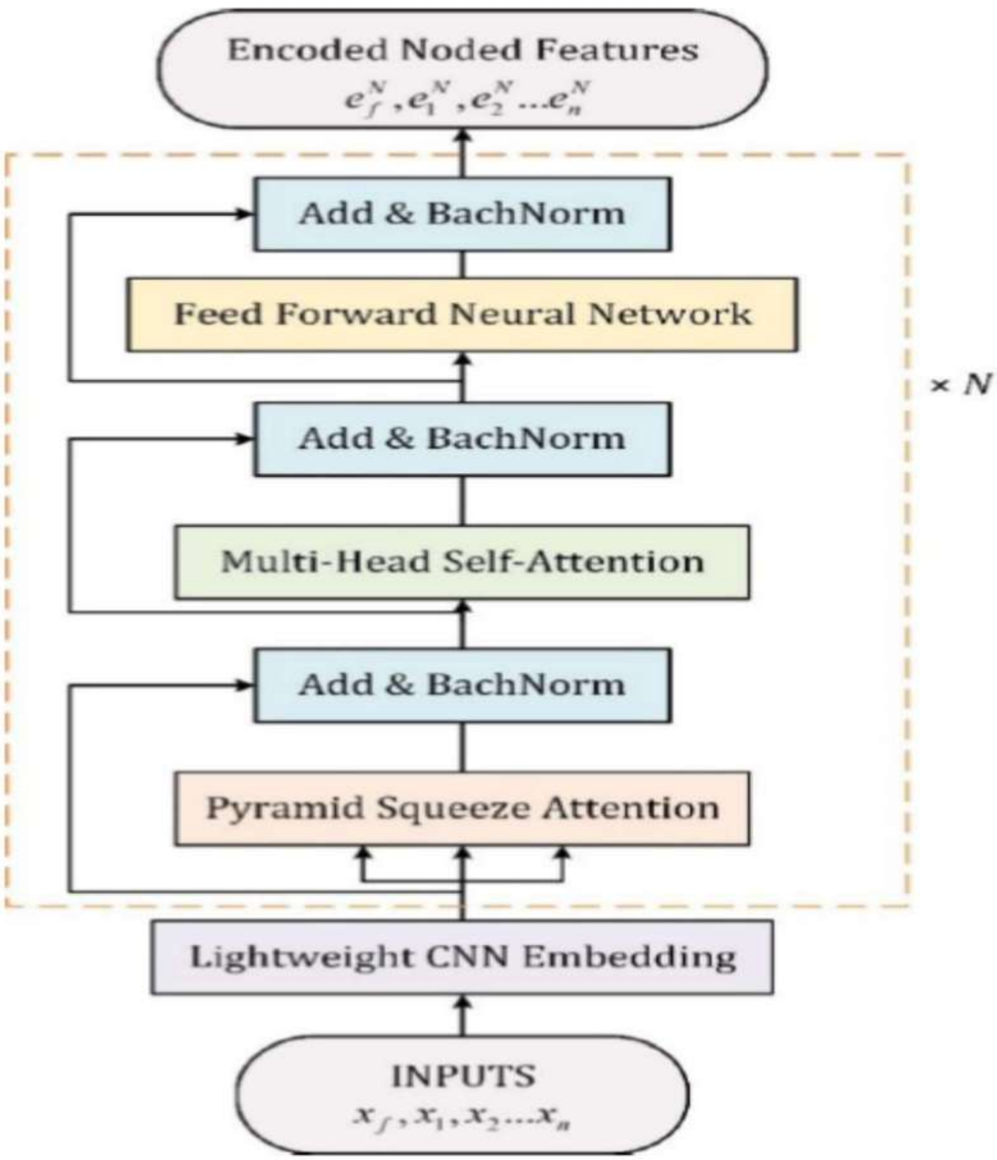
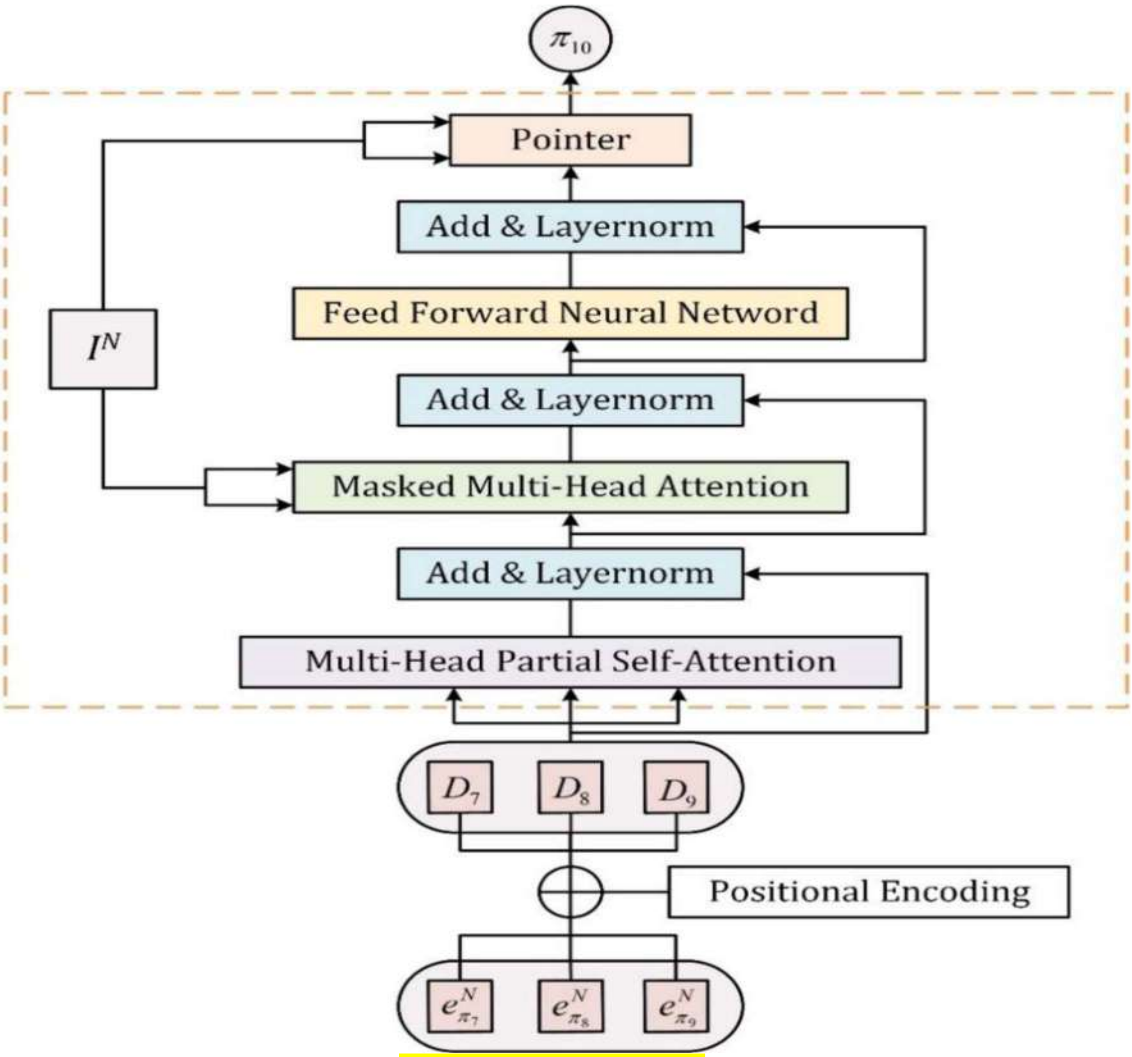


Figure 1: Overall structure of the model



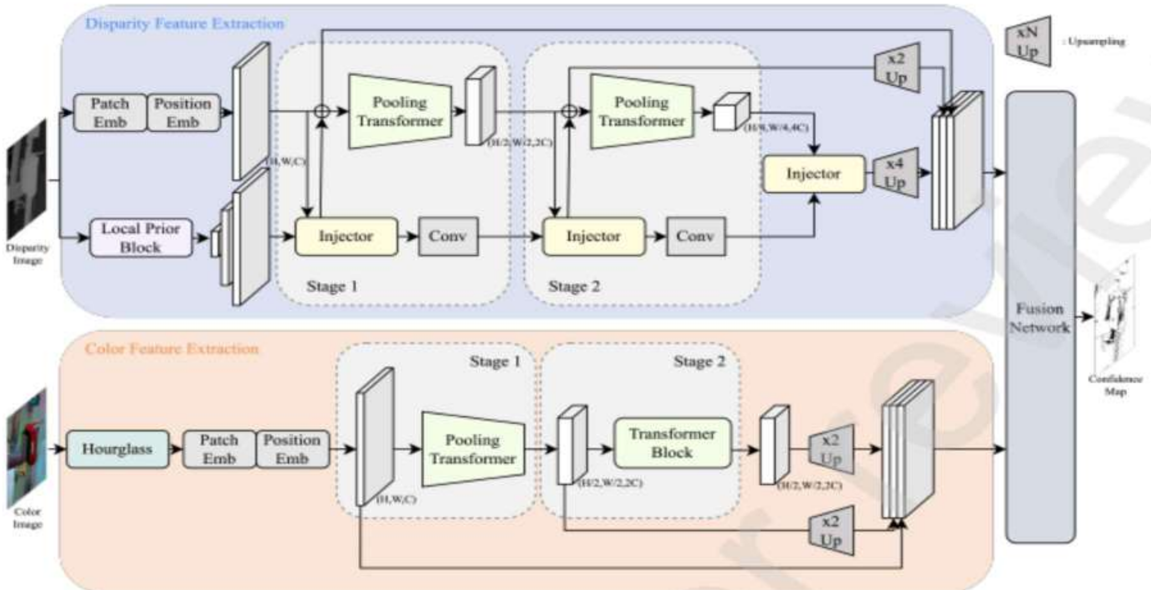
Structure of the encoder



Structure of the decoder

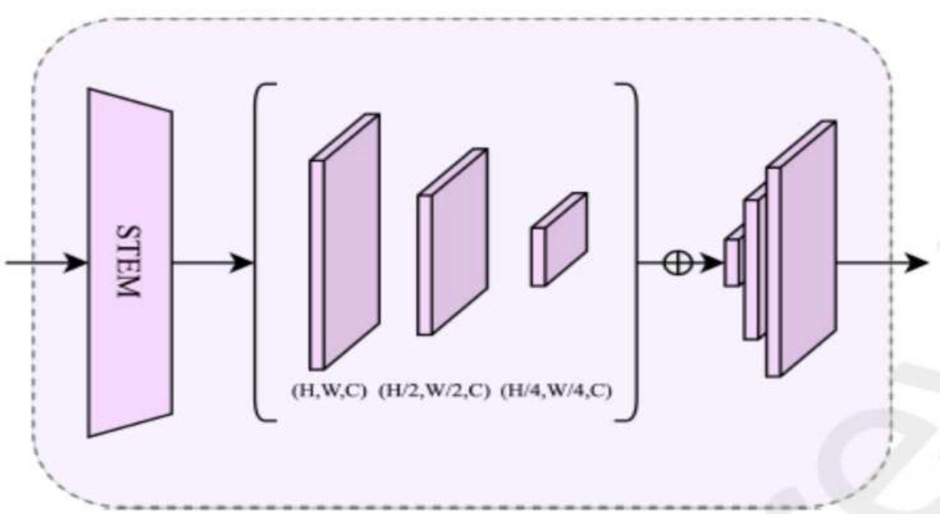
Table 1.Comparison of results of different methods on randomised TSP dataset

Method	TSP20			TSP50			TSP100		
	Len.	Gap	Time	Len.	Gap	Time	Len.	Gap	Time
Concorde[2]	3.83	0.00%	2.3m	5.69	0.00%	14m	7.76	0.00%	1.1h
LKH-3[3]	3.83	0.00%	21.2m	5.69	0.00%	27.3m	7.76	0.00%	55m
Bello et-al.*[11]	3.89	1.42%	-	5.95	4.46%	-	8.30	6.90%	-
Deudon et-al.*[12]	3.84	0.26%	-	5.81	2.07%	-	8.85	13.97%	-
Kool et-al. (greedy)*[15]	3.85	0.34%	**0s**	5.80	1.76%	2s	8.12	4.53%	**6s**
Bresson et-al. (greedy)[24]	3.89	1.57%	**0s**	5.75	1.05%	14s	8.01	3.22%	19s
Jung et-al. (greedy)[18]	**3.84**	0.25%	**0s**	5.75	0.98%	6s	8.00	3.00%	12s
Ours (greedy)	**3.84**	**0.22%**	**0s**	**5.74**	**0.95%**	**1s**	**7.95**	**2.45%**	**6s**
Kool et-al. (B=1280)*[15]	3.84	0.08%	5m	5.73	0.52%	24m	7.94	2.26%	**1h**
Bresson et-al. (B=2500)[24]	3.85	0.34%	14m	5.75	0.97%	44.8m	7.86	1.26%	1.5h
Jung et-al. (B=2500)[18]	**3.83**	**0.00%**	1.4m	5.72	0.46%	26.2m	7.86	1.22%	1.83h
Ours (B=2500)	**3.83**	**0.00%**	**1m**	**5.70**	**0.14%**	**20.4m**	**7.80**	**0.76%**	**1h**



Framework of conformer
(Convolution-augmented Transformer)

- ConFormer consists of two main branches: disparity feature extraction module and color feature extraction module. As shown in the name, the disparity map and color image are input for each module.



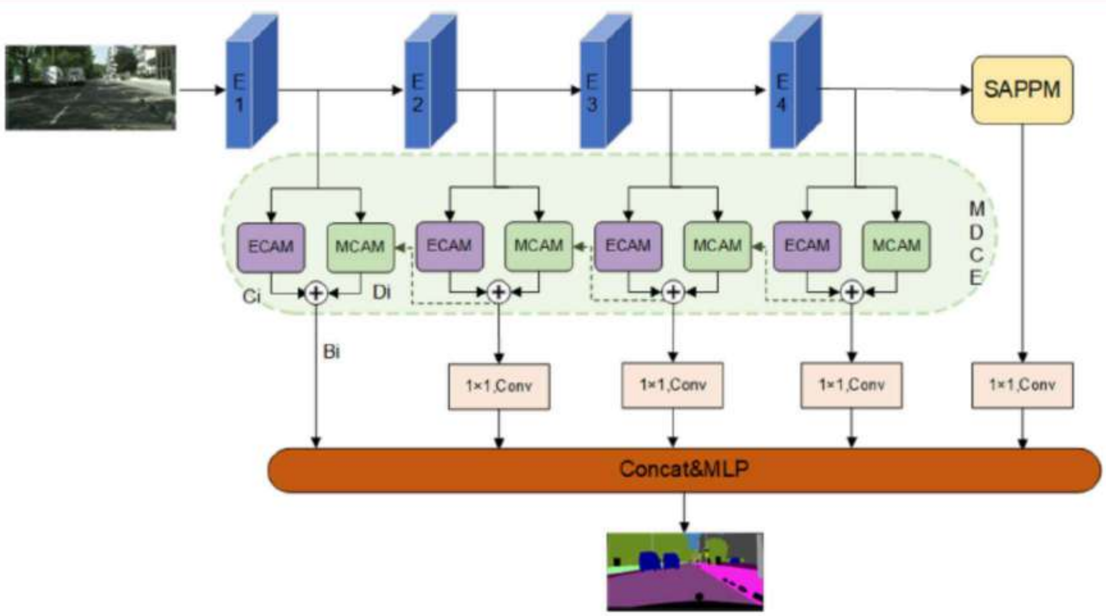
Specific network of local prior block

The average AUC values for FlyingThings, Driving, KITTI 2012, KITTI 2015 dataset by using disparity only as input.

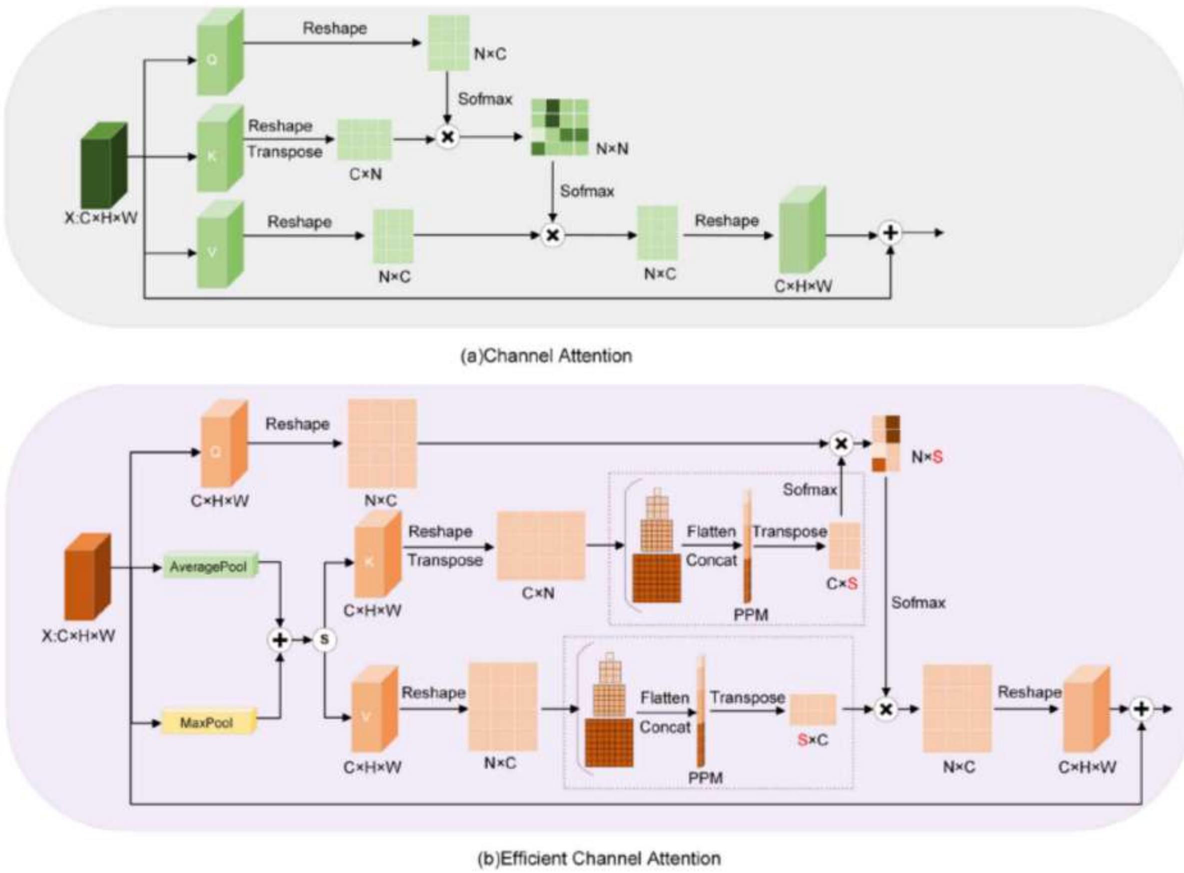
✓ The AUC value of ground truth confidence is measured as 'Optimal'. The result with the lowestAUC value in each experiment is highlighted.

Methods	FlyingThings [25]		Driving [25]		KITTI 2012 [26]		KITTI 2015 [27]	
	PSM	GA-Net	PSM	GA-Net	PSM	GA-Net	PSM	GA-Net
CCNN-S [9]	9.783	3.598	13.150	5.047	0.079	0.906	0.015	0.039
ConfNet-S [2]	10.060	3.719	13.502	4.898	0.040	1.078	0.012	0.062
LGC-Net-S [2]	10.272	4.185	12.822	5.313	0.041	0.818	0.017	0.039
LAF-Net-S [11]	8.372	3.350	12.260	4.358	0.032	0.575	0.014	0.035
ConFormer-S	**7.726**	**2.963**	**10.784**	**3.168**	**0.029**	**0.339**	**0.011**	**0.023**
Optimal	4.502	1.602	4.040	1.399	0.002	0.086	0.0002	0.0004

Transformer Net	**2025-45**	

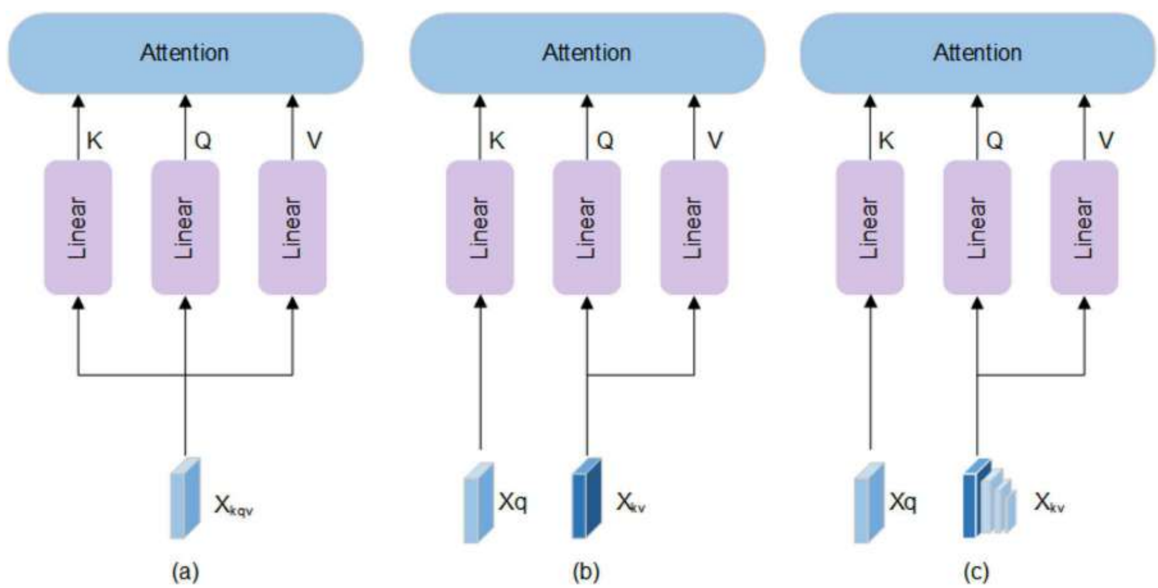


Overview of the proposed MEMAFormer



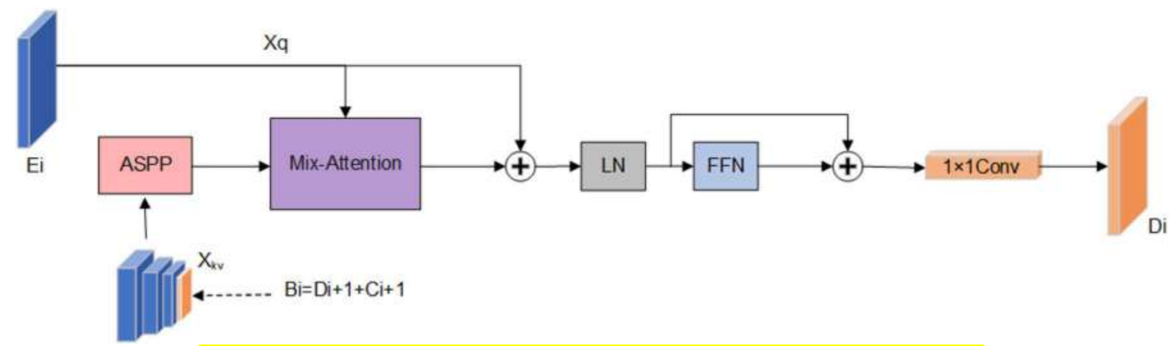
Comparison of attention frames of the channel attention
(a) efficient channel attention (b), where S << N, N _ H × W.

- ✓ **Symbols "⊗", "⊕" and "sig " represent, addition, elementwise multiplication, and the sigmod function, respectively**

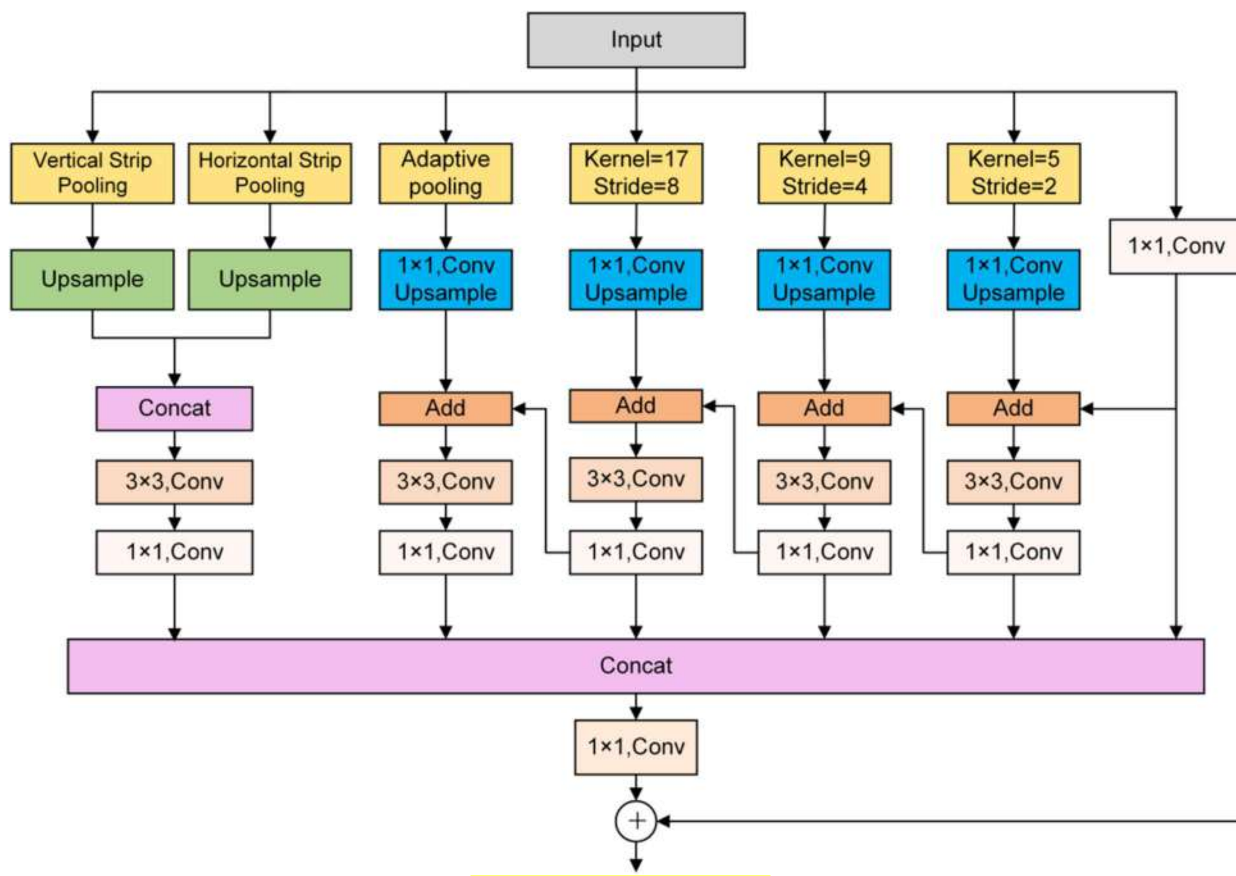


Comparison of three different attentions

- ✓ **a self-attention, b cross-attention, c mix-attention**



The overall architecture of multi-layers cross attention module



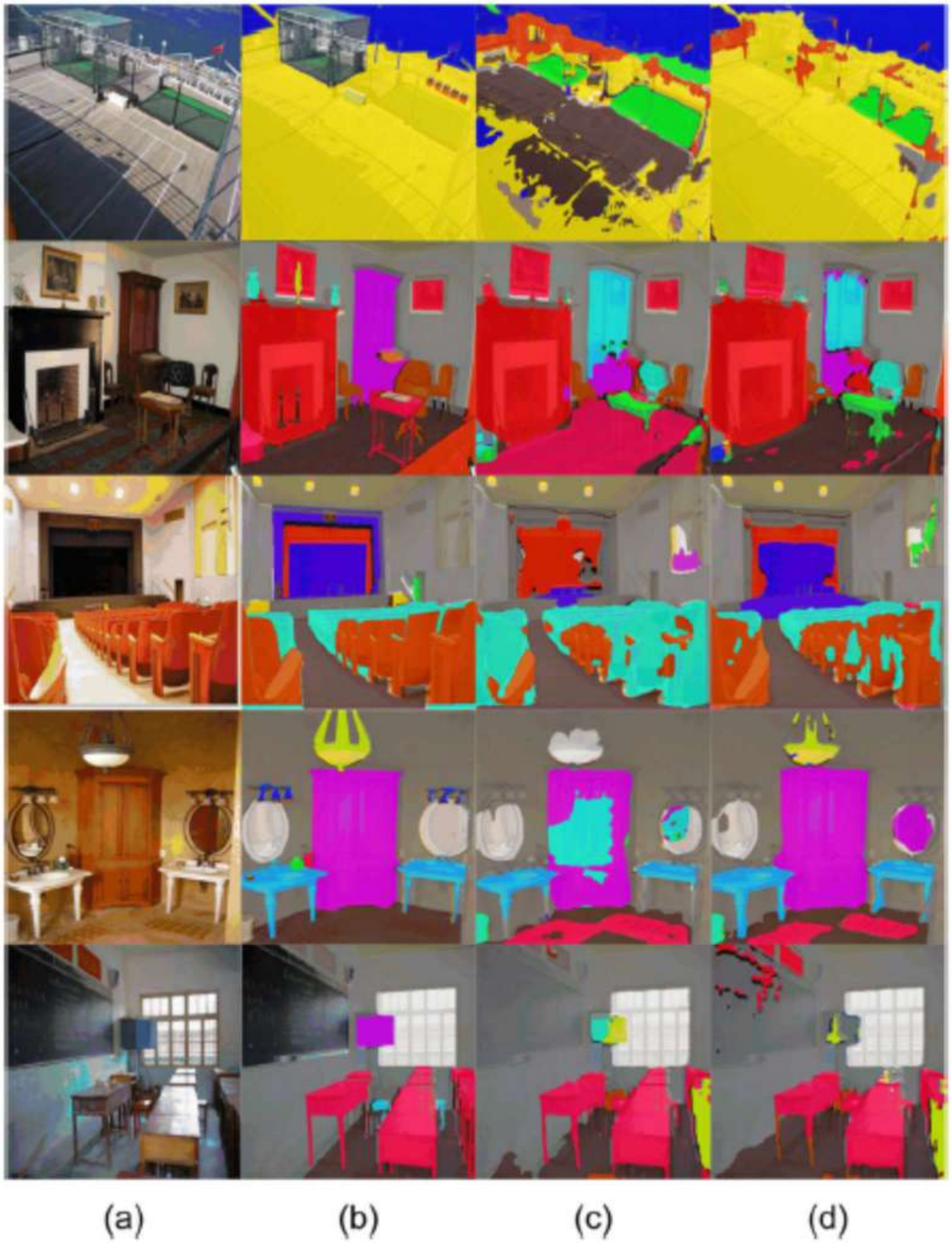
Architecture of SAPPM

Method	Backbone	Parameters (M)	mIou	
			ADE20K	Cityscapes
SETR-PUP (80 k) [25]	ViT-B	98	46.3	–
	ViT-L	**318**	48.6	**82.15**
Swin transformer [32]	Swin-T	60	46.1	–
	Swin-S	81	**49.3**	–
PVT [50]	PVT v2-B0	7.6	37.2	–
	PVT v2-B1	18	42.6	–
Segmenter [60]	Seg-B	86	48.1	80.5
	Seg-B/Mask	**86**	**50.1**	80.6
HRFormer (150k) [67]	HRFormer-S	14	44.0	80.0
	HRFormer-B	50	46.3	81.4
MaskFormer [68]	Swin-T	42	46.7	78.5
	Swin-S	63	**49.8**	–
Mask2Former [69]	Swin-T	47	47.7	**82.1**
	Swin-B	**216**	–	**83.3**
Segformer [26]	MiT-B0	3.8	37.4	76.2
	MiT-B1	13.7	42.2	78.5

Ablation experiments of transformer variants and MEMAFormer

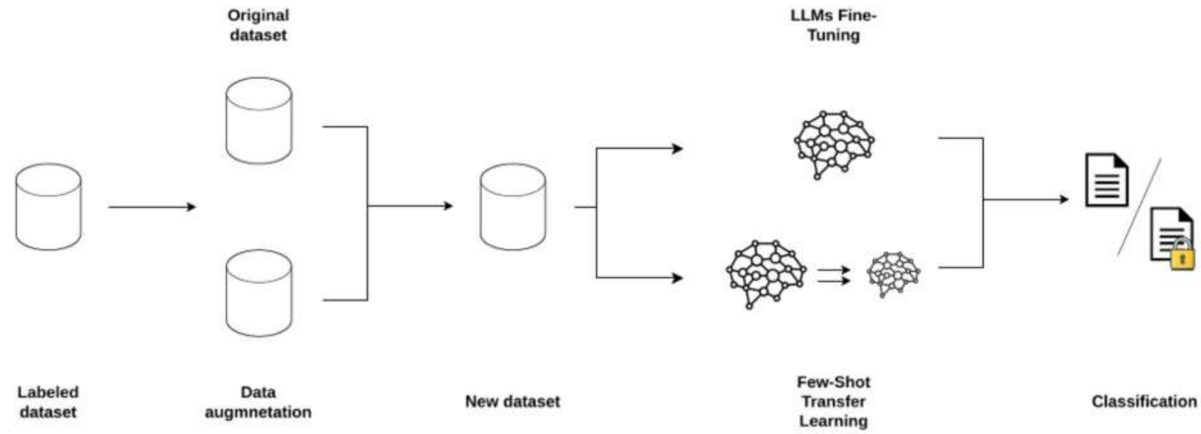
Method	Backbone	Parameters (M)	GFLOPs	FPS	mIou
FCN [20]	MobileNetV2	9.8	39.6	64.4	19.7
PSPNet [53]	MobileNetV2	13.7	52.9	57.7	29.7
DeepLabV3+ [61]	MobileNetV2	15.4	69.4	43.1	34.0
Segformer [26]	MiT-B0	3.8	8.4	50.5	37.4
FeedFormer [62]	MiT-B0	4.6	7.8	79.5	39.2
U-MixFormer [66]	MiT-B0	6.1	9.1	55.4	41.2
MEMAFormer	MiT-B0	6.4	14.3	49.4	42.2

Ablation study showing GFLOPs, parameters and FPS on ADE20K



Visual comparison on ADE20K Dataset.

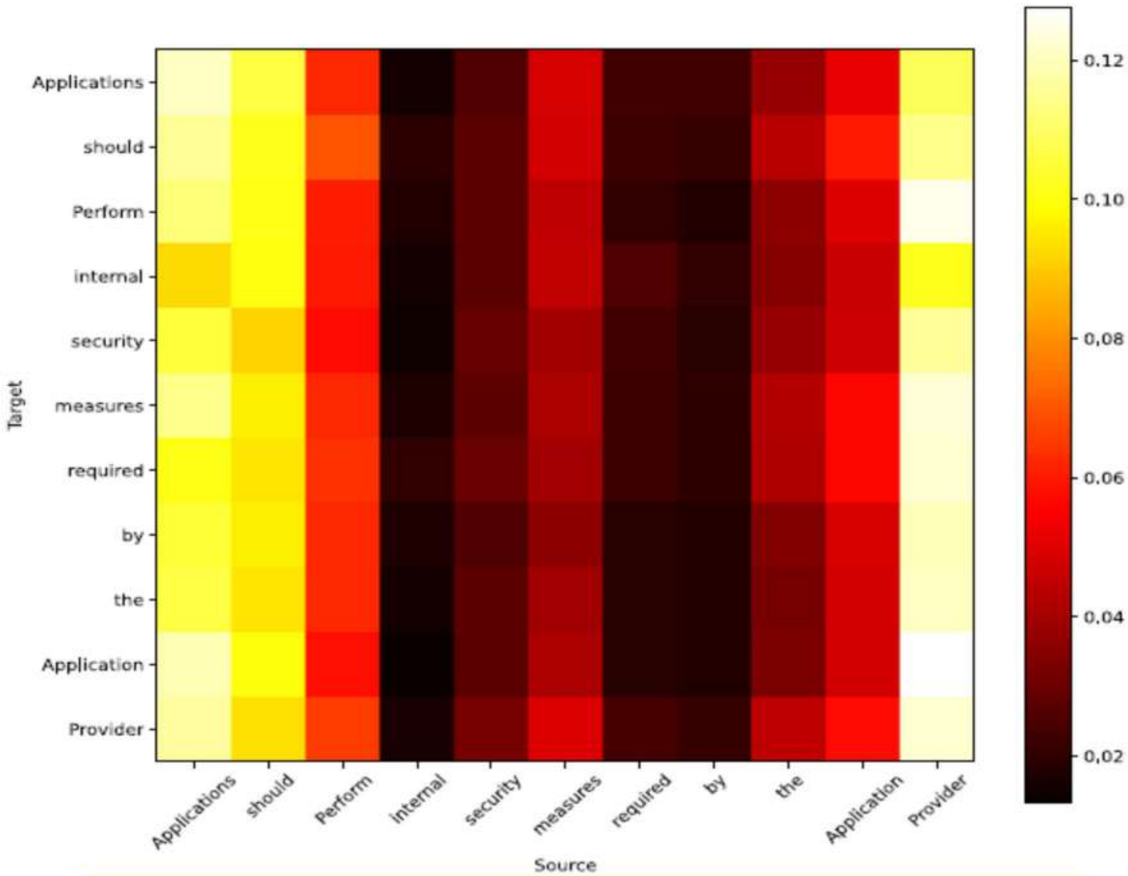
- ✓ a Image. b Ground truth. c Segformer. d Ours



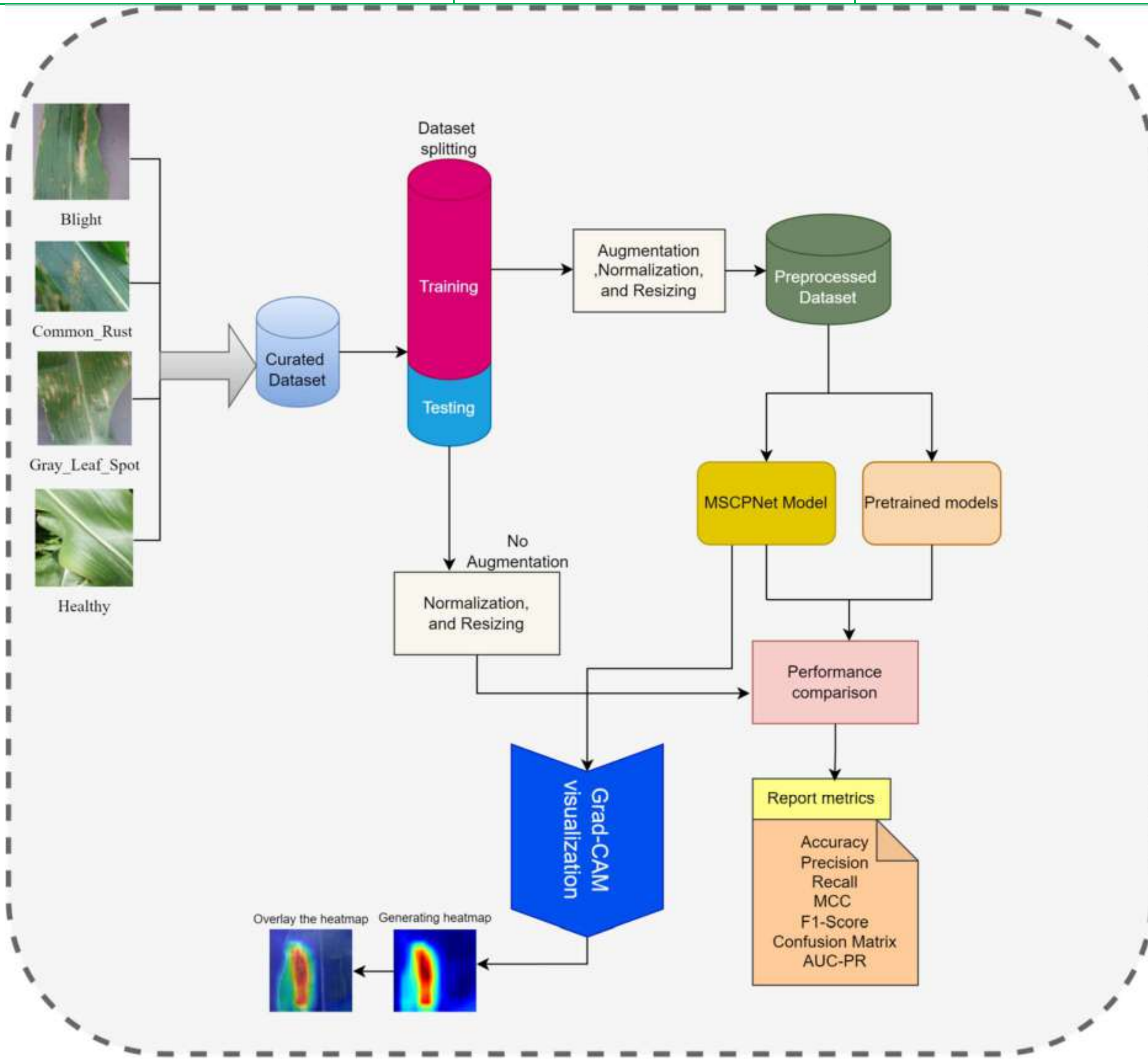
The workflow of proposed method to classify security and non-security requirements that rely on LLMs fine-tuning and Few-Shot learning with transfer learning models leveraging pre-trained models using data augmentation

Table 5. Results of the fine-tuned models in terms of accuracy, precision, recall, and F1-score on the four datasets.

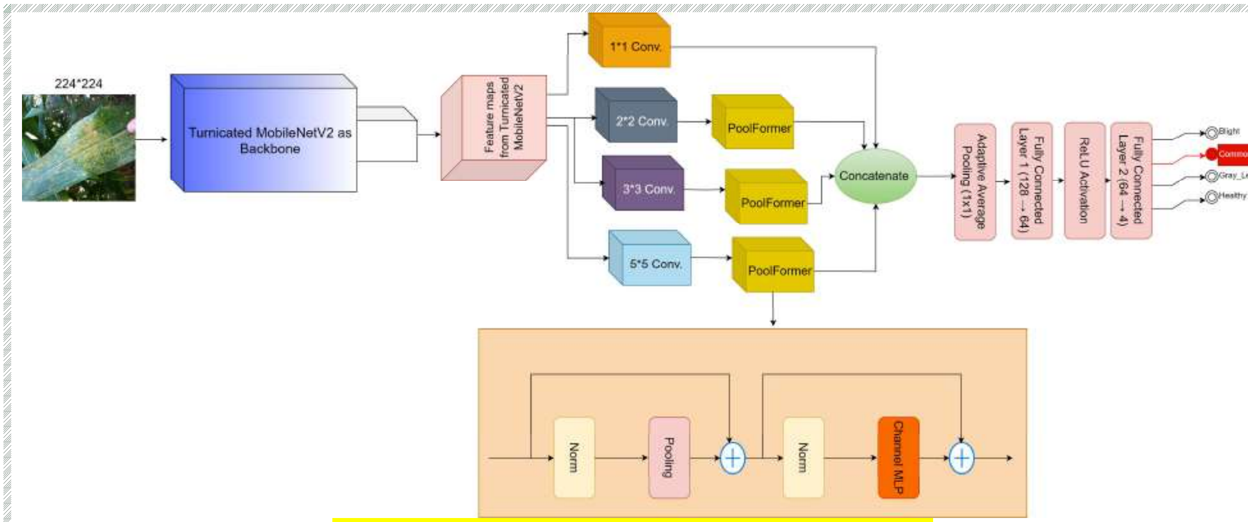
Model	Accuracy	Precision	Recall	F1-Score
Normal dataset				
BERT-based-uncased	0.89	0.89	0.76	0.82
DistilBERT-base-uncased	0.89	0.89	0.86	0.82
RoBERTa-base	0.9	0.86	0.86	0.86
xlnet-base-uncased	0.9	0.89	0.81	0.85
Contextual word augmented dataset				
BERT-based-uncased	0.94	0.9	0.9	0.9
DistilBERT-base-uncased	0.89	0.85	0.81	0.83
RoBERTa-base	0.84	0.92	0.57	0.71
xlnet-base-uncased	0.92	0.94	0.81	0.87
Back translation augmented dataset				
BERT-based-uncased	0.92	0.9	0.96	0.88
DistilBERT-base-uncased	0.92	0.9	0.96	0.88
RoBERTa-base	0.84	0.87	0.62	0.71
xlnet-base-uncased	0.9	0.83	0.9	0.86



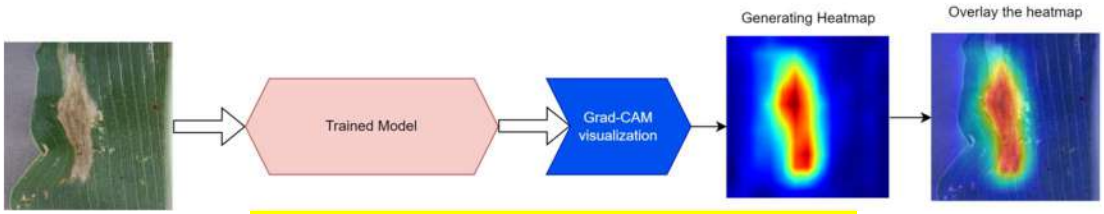
Heatmap showing the attention visualization of the last layer (12th layer) of the Roberta-base model fine-tuned in the original dataset



Flowchart of the complete methodology



MSCPNet model for maize disease classification



Grad-CAM visualization of MSCPNet model for maize disease classification

Algorithm 2: MSCPNet for Maize Disease Classification

Input: $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$: Input maize leaf image
Output: $\hat{y} \in \mathbb{R}^{\text{num\_classes}}$: Predicted class probabilities

Step 1: Feature Extraction

- Load pre-trained MobileNetV2 and truncate after a certain layer.
- Extract base feature map:

$$\mathbf{F}_b \leftarrow \text{MobileNetV2}(\mathbf{X})$$

Step 2: Channel Reduction

- Apply 1×1 convolution to reduce feature map dimensions:

$$\mathbf{F}_r \leftarrow \text{Conv}_{1\times 1}(\mathbf{F}_b)$$

Step 3: Multi-Scale Convolution and PoolFormer Application

- Apply multi-scale convolutions and PoolFormer blocks:

$$\mathbf{F}_k \leftarrow \mathcal{P}_k(\text{Conv}_{k\times k}(\mathbf{F}_r)), \quad k \in \{2, 3, 5\}$$

- Apply 1×1 convolution branch in parallel:

$$\mathbf{F}_{1x1} \leftarrow \text{Conv}_{1\times 1}(\mathbf{F}_r)$$

Step 4: Feature Concatenation

- Concatenate the outputs from all convolution branches:

$$\mathbf{F}_{\text{concat}} \leftarrow \text{Concat}(\mathbf{F}_2, \mathbf{F}_3, \mathbf{F}_5, \mathbf{F}_{1x1})$$

Step 5: Apply 1 × 1 convolution and Layer Normalization

- Apply 1×1 convolution to reduce dimensions of concatenated features:

$$\mathbf{F}_{\text{final}} \leftarrow \text{Conv}_{1\times1}(\mathbf{F}_{\text{concat}})$$

- Apply layer normalization:

$$\mathbf{F}_{\text{enh}} \leftarrow \text{LayerNorm}(\mathbf{F}_{\text{final}})$$

Step 6: Global Average Pooling and Classification

- Perform global average pooling on the normalized features:

$$\mathbf{F}_{\text{pool}} \leftarrow \text{GAP}(\mathbf{F}_{\text{enh}})$$

- Flatten the pooled features and pass through fully connected layers:

$$\mathbf{z}_1 \leftarrow \text{ReLU}(\mathbf{W}_1\mathbf{F}_{\text{pool}}), \quad \hat{y} \leftarrow \mathbf{W}_2\mathbf{z}_1$$

Output: Return predicted class probabilities $\hat{y}$.

TABLE 4: Hyperparameter configurations

Hyperparameter	Configurations
Input Size	224
Optimizers	Adagrad AdamW SGD Adam RMSprop
Batch Size	8, 16, 32
Learning Rate	0.01, 0.001, 0.0001
Early Stopping	Patience of 10 epochs
Loss Function	Categorical Cross Entropy
Number of Epochs	200

Performance comparison of different pre-trained models, truncated MobileNetV2, and the proposed MSCPNet model.

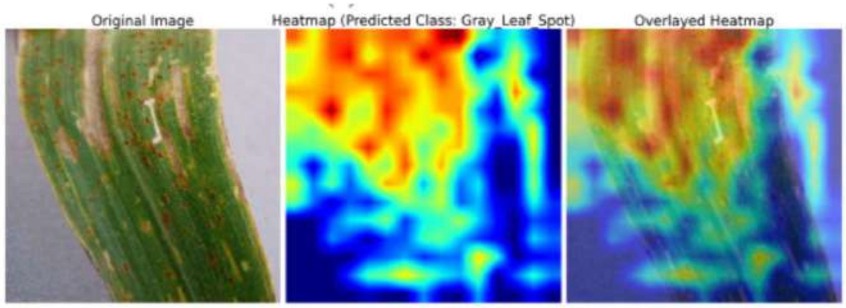
Model	Accuracy	Precision	Recall	F1-Score	MCC	Inference Time (s)
DenseNet121	95.53%	94.37%	95.15%	94.69%	0.9396	0.0289
ResNet50	95.21%	93.71%	95.41%	94.37%	0.9356	0.0120
ShuffleNetV2	94.09%	92.20%	93.98%	92.82%	0.9209	0.0107
SqueezeNet	93.61%	92.13%	92.25%	92.12%	0.9135	0.0091
MobileNetV2	96.65%	96.26%	95.85%	96.04%	0.9542	0.0110
Truncated MobileNetV2	95.85%	95.02%	95.19%	95.09%	0.9434	0.0083
Proposed MSCPNet	**97.44%**	**96.76%**	**97.37%**	**97.04%**	**0.9653**	**0.0111**

Performance Comparison of different backbone models with the Proposed MSCPNet block

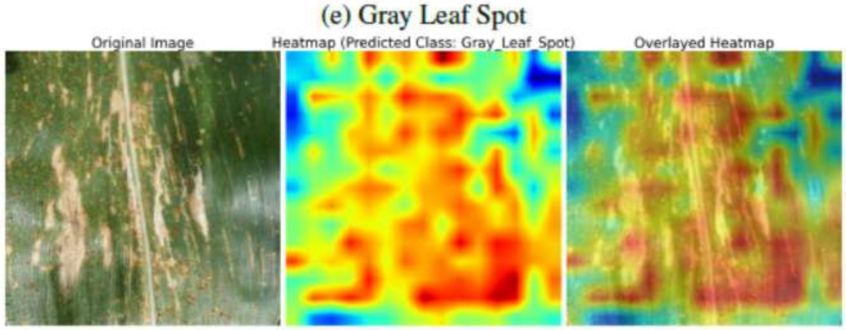
Backbone Model	Accuracy	Precision	Recall	F1-Score	MCC	Inference Time (s)	Total FLOPs
DenseNet121	95.85%	94.82%	95.76%	95.24%	0.9436	0.0302	3,066,045,440
ResNet50	95.53%	94.16%	94.42%	94.29%	0.9390	0.0180	4,915,031,552
ShuffleNetV2	95.53%	93.96%	95.31%	94.51%	0.9396	0.0150	318,950,600
SqueezeNet	94.41%	92.84%	93.78%	93.25%	0.9241	0.0136	1,385,824,800
MobileNetV2	97.28%	96.37%	97.03%	96.67%	0.9631	0.0181	517,634,624
Proposed MSCPNet	**97.44%**	**96.76%**	**97.37%**	**97.04%**	**0.9653**	**0.0111**	**315,258,752**

TABLE 7: Performance comparison with different optimizers.

Optimizer	Accuracy	Precision	Recall	F1-Score	MCC
Adagrad	96.33%	95.62%	94.96%	95.26%	0.9498
Adam	**97.44%**	**96.76%**	**97.37%**	**97.04%**	**0.9653**
AdamW	96.33%	95.44%	95.75%	95.58%	0.9500
RMSprop	96.49%	95.84%	95.90%	95.87%	0.9521
SGD	97.12%	96.83%	96.16%	96.45%	0.9609

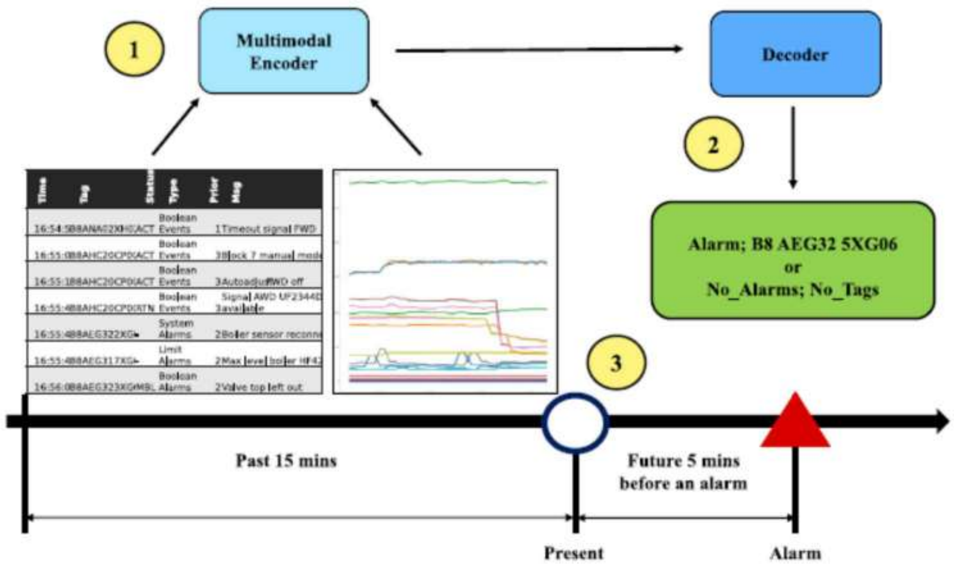


(e) Gray Leaf Spot



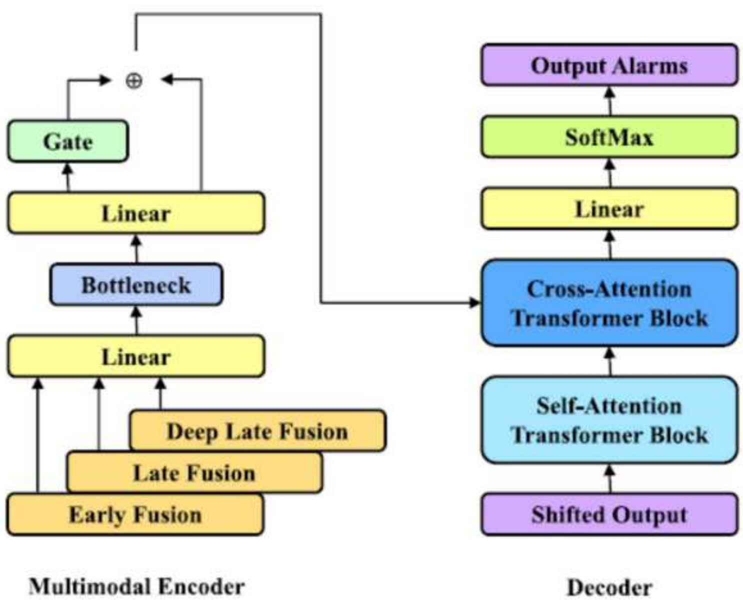
(f) Gray Leaf Spot

Grad-CAM visualizations for different disease classes



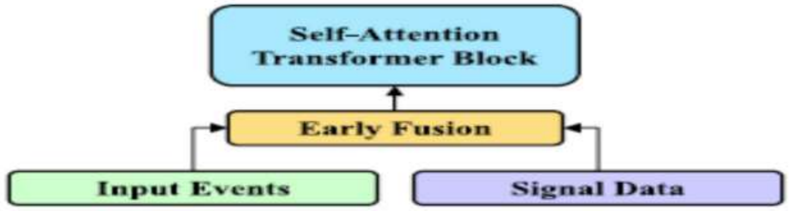
Input and output modalities:

- ✓ The encoder creates a representation of the past 15 min of events and signal data, which is used by decoder to forecast an alarm 5 min into the future.

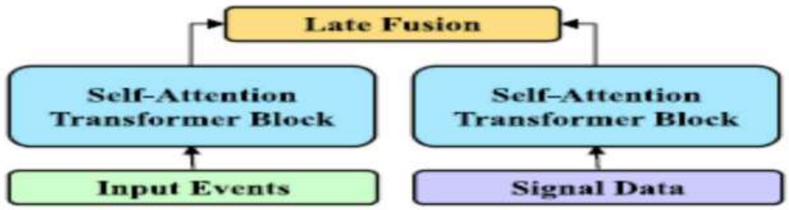


MUST Architecture.

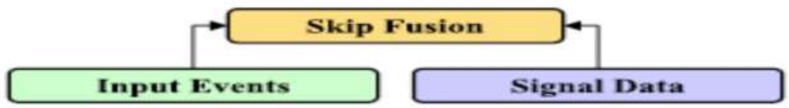
- ✓ **Encoder incorporates multiple fusion layers, concatenated and**
- ✓ **Passed through a bottleneck layer guided by a sigmoid gate**
- ✓ **The resulting representation is passed to the cross-attention block of the decoder**



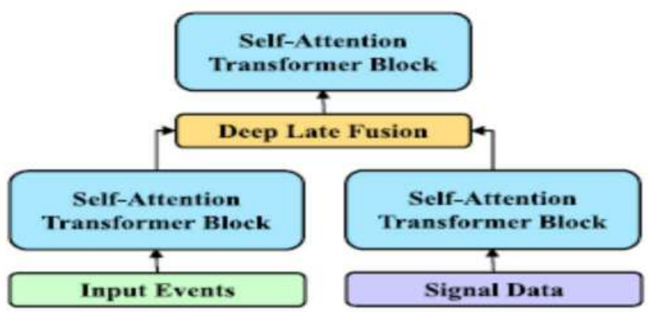
(a) Early Fusion



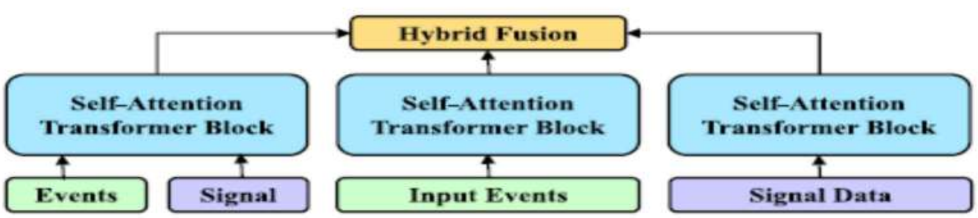
(b) Late Fusion



(c) Skip Fusion (ours, only as baseline)



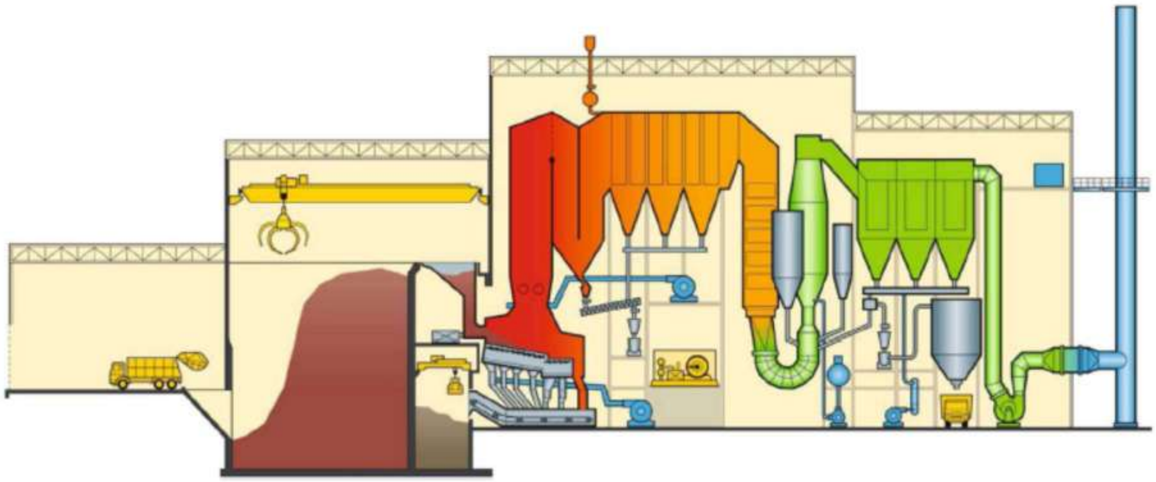
(d) Deep Late Fusion (ours)



(e) Hybrid fusion (ours, only as baseline)

Fusion types:

- ✓ State-of-the-art early and late fusion and the novel skip, deep late and hybrid fusion.
- ✓ Only early, late and deep late fusion are used as building blocks in MultiFusion Transformer.
- ✓ All the fusion types are used as baselines



Example of a waste incineration plant scheme

- **Waste is stored and mixed in the reception area.**
- **From there, it is moved on the grate to the combustion chamber where it is burnt using flue gas and pre-heated air, and**
- **Resulting steam is used to generate electricity, yet it can also be used directly, e. g., for district heating.**
- **Finally, gases are cleaned before being emitted from the plant.**

Table 3
Performance of MultiFusion compared to SotA fusion techniques based on experiments on the real dataset, plant B (best in **bold**).

	F1 Bin ↑	F1 MC ↑	BLEU ↑	Params (M)	Time
Skip Fusion	84.1 ± 0.2	41.2 ± 0.2	37.8 ± 1.0	1.6 ± 0.0	1.8 ± 0.4
Early Fusion	84.1 ± 0.3	41.9 ± 0.5	39.2 ± 1.0	1.9 ± 0.0	4.2 ± 0.9
Late Fusion	84.0 ± 0.3	42.5 ± 0.2	39.4 ± 1.2	2.2 ± 0.0	6.2 ± 1.3
Deep Late	84.0 ± 0.4	42.5 ± 0.5	39.6 ± 1.4	2.5 ± 0.0	7.9 ± 2.1
Hybrid Fusion	84.2 ± 0.3	42.2 ± 0.3	39.2 ± 0.5	2.5 ± 0.0	7.9 ± 1.4
MultiFusion	**84.3 ± 0.3**	**42.7 ± 0.8**	**40.4 ± 1.5**	2.9 ± 0.0	4.1 ± 1.0

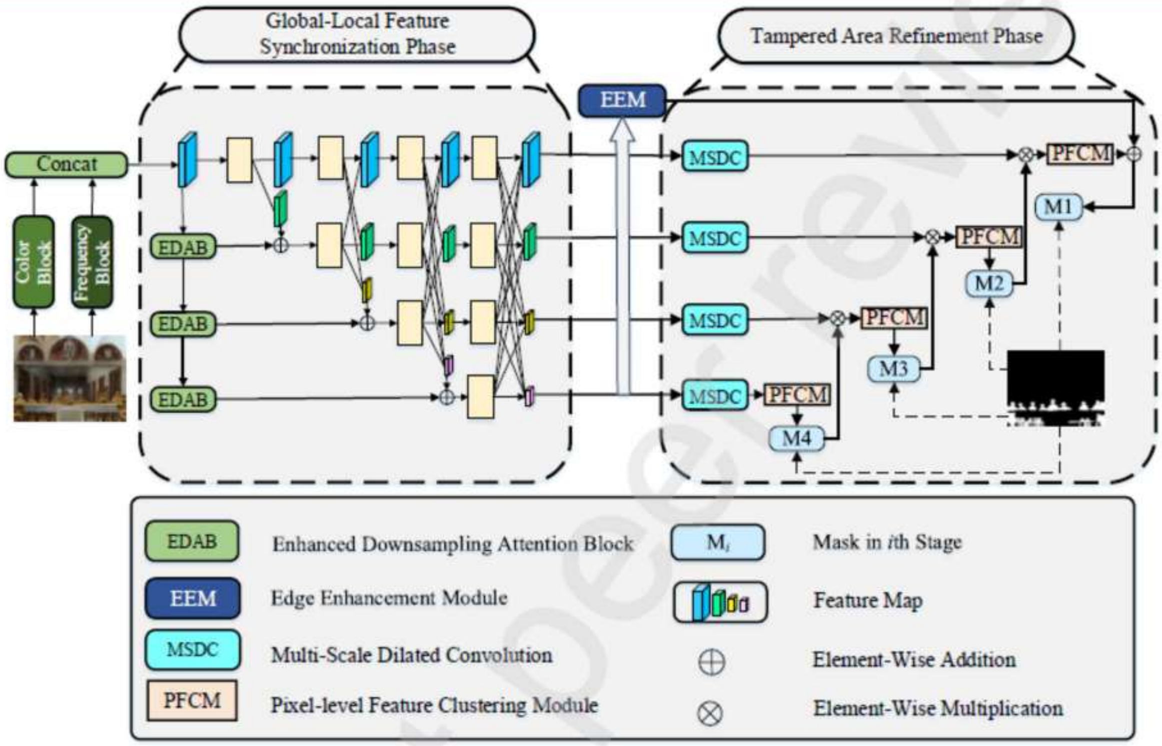


Fig. 2. Structure of Multi-scale Iterative Tamper Detection Net.

- The network primarily consists of the global-local feature synchronization phase and the tampered area refinement phase.
- During the global-local feature synchronization stage, the adoption of a parallel network architecture ensures that the network can simultaneously capture feature information at different scales.
- The Enhanced Downsampling Attention Block (EDAB) is introduced to utilize the image's multi-level features, providing sufficient spatial information for each layer in the network and maintaining close connections between different scales, thereby effectively addressing the issue of scale variations.
- Accurate prediction of tampering masks is achieved through feature fusion, ensuring the comprehensive utilization of features at different scales in tampering detection.
- Additionally, boundary information is processed through the Edge Enhancement Module (EEM) edge supervision branch, which serves as an auxiliary to the main feature extraction process

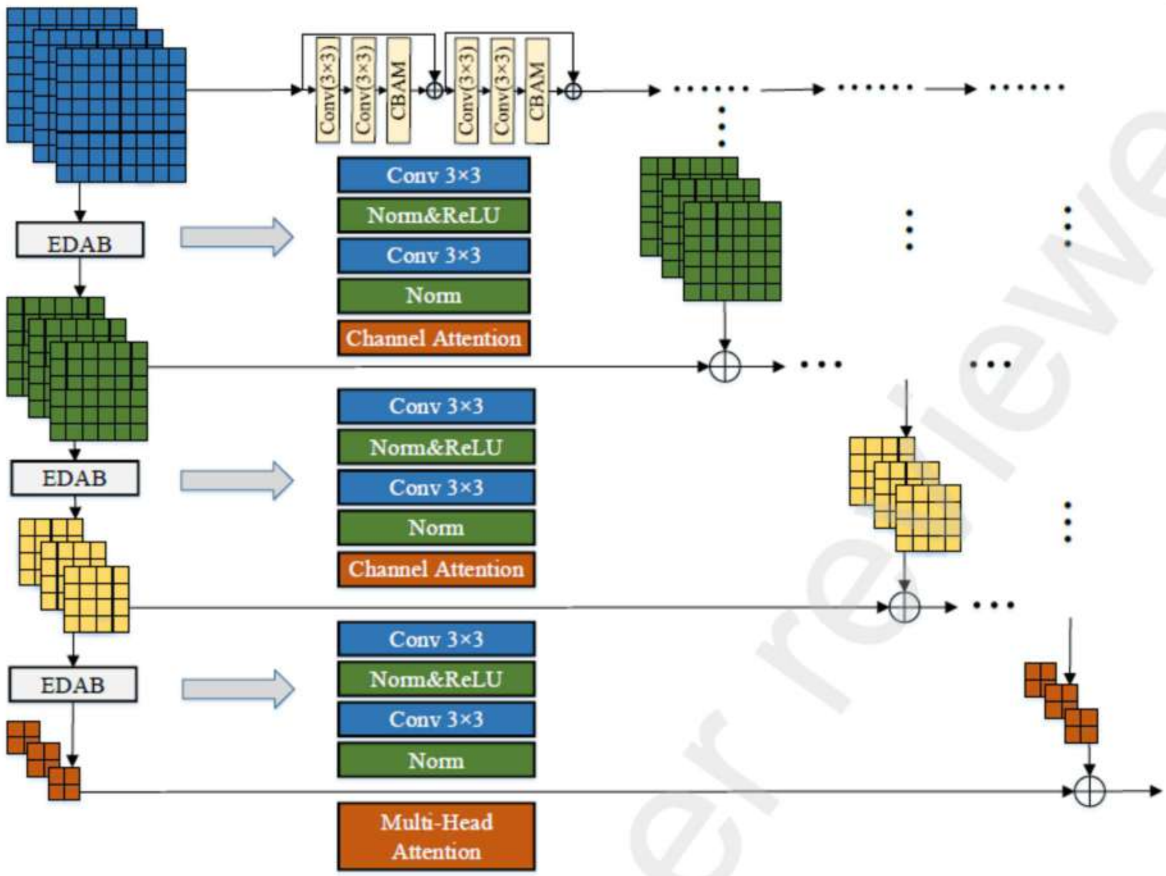


Fig. 3. Structure of Enhanced Downsampling Attention Block (EDAB).

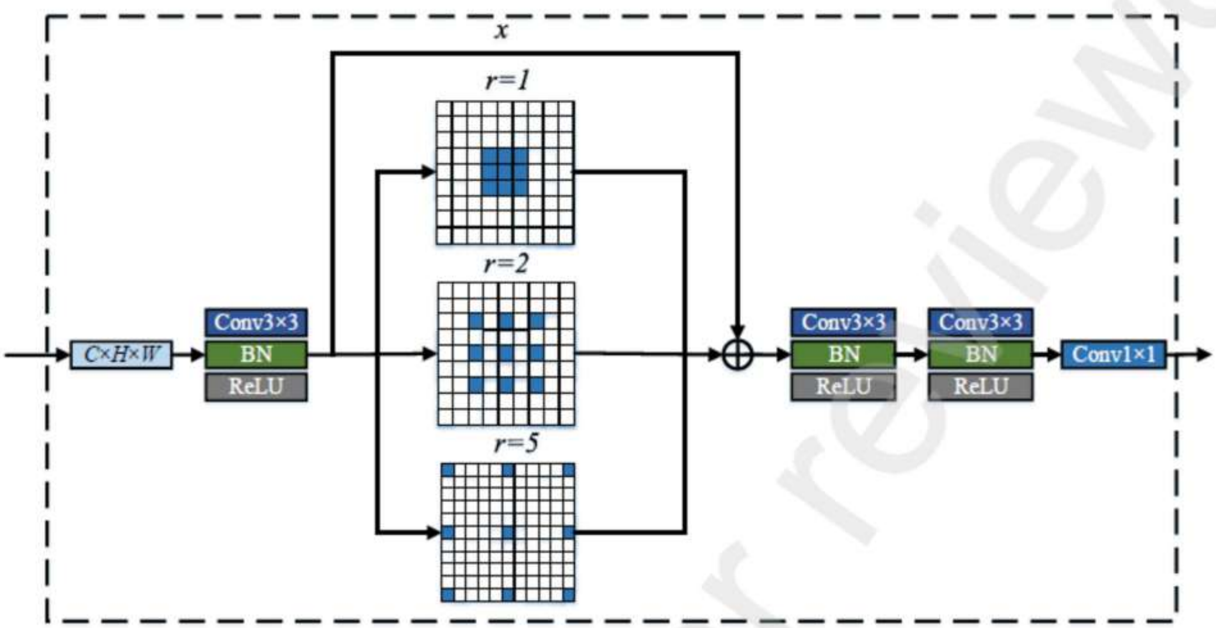


Fig. 4. Multi-Scale Dilated Convolution (MSDC) structure diagram.

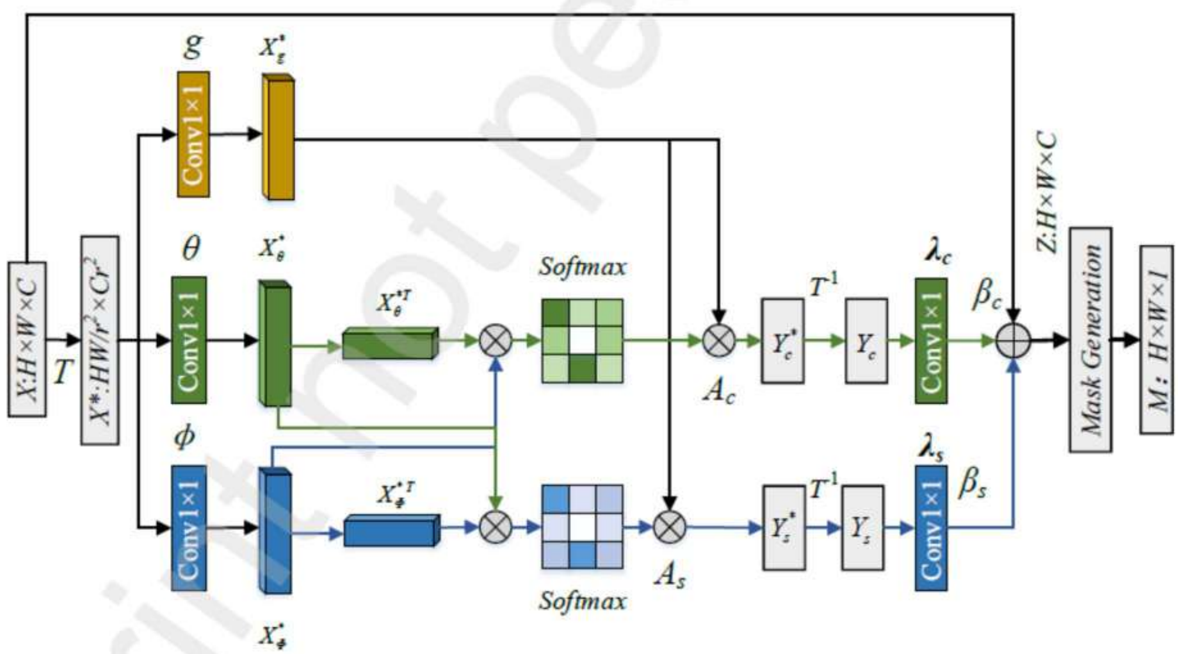


Fig. 5. Structure of Pixel-level Feature Clustering Module (PFCM).

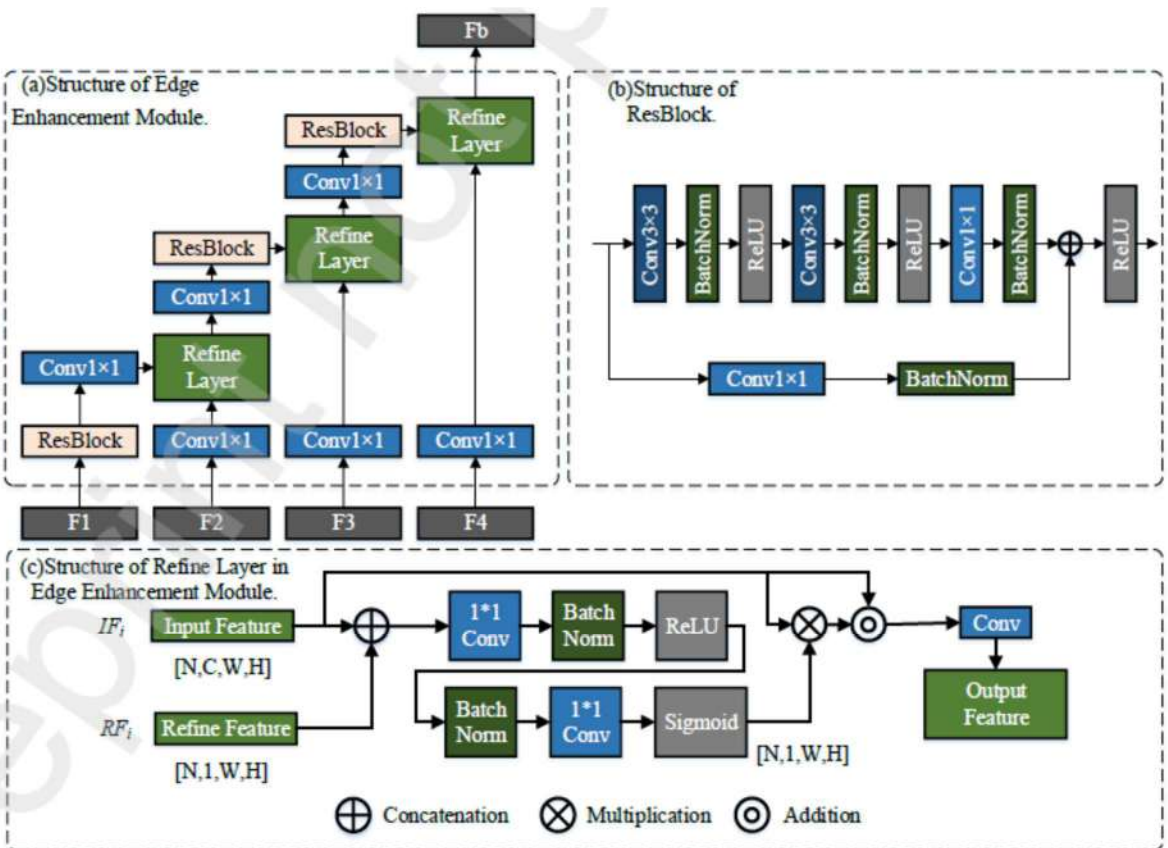
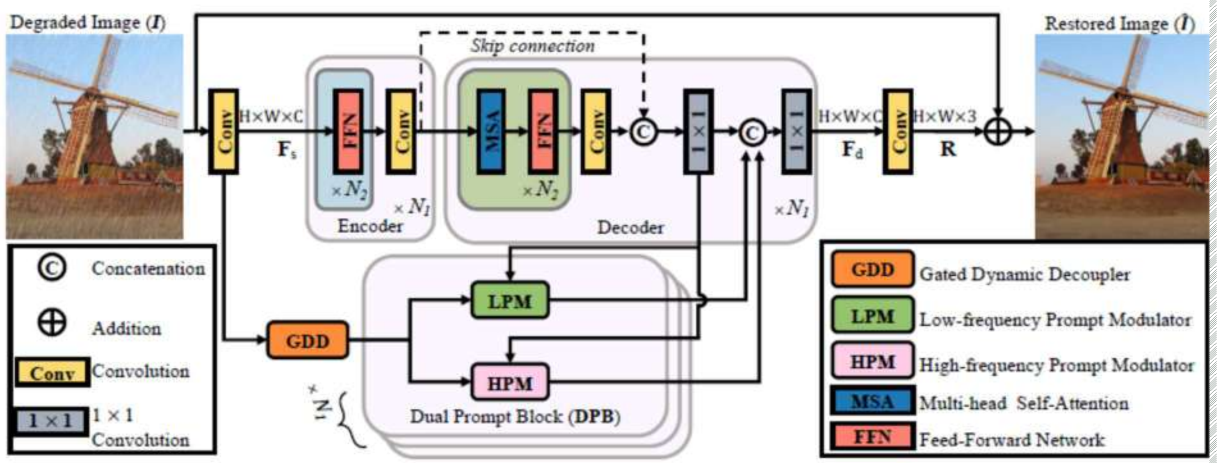


Fig. 6. Schematic diagram of the Edge Enhancement Module.



Overview of FPro.

- Except for the common upper restoration branch, which is similar to existing methods, FPro contains another bottom prompt branch to extract informative features from a frequency perspective.
- Specifically, the primary components of the prompt branch in this framework are the gated dynamic decoupler (GDD) and dual prompt block (DPB).
- The GDD is employed to decompose the low-frequency components and corresponding high-frequency characteristics from the input features.
- Then these frequency-specific features are further processed in DPB, i.e., the high-frequency prompt modulator (HPM) and the lowfrequency prompt modulator (LPM), which generates representative frequency prompt to facilitate the clear image reconstruction.

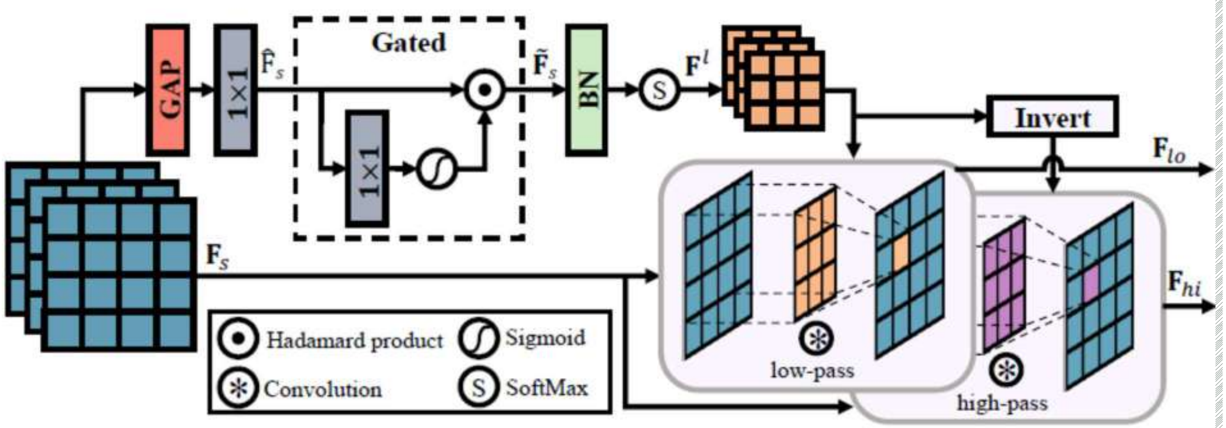
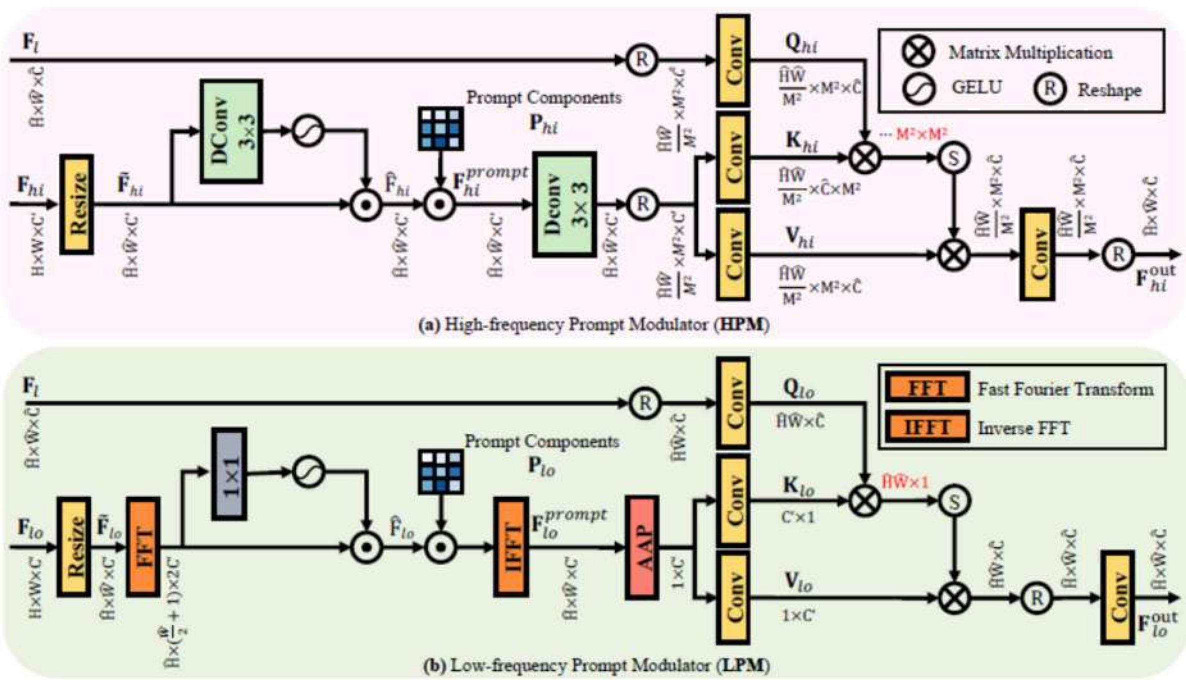


Fig. 2: Illustrations of the Gated Dynamic Decoupler.



Illustrations of the proposed components. (a) High-frequency Prompt Modulator (HPM); (b) Low-frequency Prompt Modulator (LPM

Table 1: Quantitative comparison on SPAD [62] for rain streak removal.

Method	SPAD [62] PSNR ↑	SSIM ↑
DDN [13]	36.16	0.9463
PReNet [50]	40.16	0.9816
RCDNet [61]	43.36	0.9831
MPRNet [75]	43.64	0.9844
SPAIR [46]	44.10	0.9872
Uformer-S [64]	46.13	0.9913
SCD-Former [15]	46.89	**0.9941**
IDT [67]	47.34	0.9929
Restormer [73]	47.98	0.9921
DRSformer [5]	<u>48.53</u>	0.9924
FPro (Ours)	**48.99**	<u>0.9936</u>

Table 2: Quantitative comparison on AGAN-Data [47] for raindrop removal.

Method	AGAN-Data [47] PSNR ↑	SSIM ↑
Eigen's [12]	21.31	0.757
Pix2pix [20]	28.02	0.855
Uformer-S [64]	29.42	0.906
WeatherDiff$_{128}$ [42]	29.66	0.923
TransWeather [55]	30.17	0.916
DuRN [33]	31.24	0.926
RaindropAttn [48]	31.37	0.918
AttentiveGAN [47]	31.59	0.917
IDT [67]	31.63	<u>0.936</u>
Restormer [73]	<u>31.68</u>	0.934
FPro (Ours)	**31.96**	**0.937**