Wave-wise Discriminative Tracking by Phase-Amplitude Separation, Augmentation and Mixture

Huibin Tan[1] *, **Mingyu Cao**[1] *, **Kun Hu**[2], **Xihuai He**[1], **Zhe Wang**[3],
Hao Li[1], **Long Lan**[1] † and **Mengzhu Wang**[4] †
[1]College of Computer Science and Technology, National University of Defense Technology
[2]Independent Researcher
[3]Hong Kong Polytechnic University
[4]Hebei University of Technology
{tanhb_, caomy720}@nudt.edu.cn, hu_kun_@outlook.com, hexihuai23@nudt.edu.cn,
zhe-leo.wang@connect.polyu.hk, leemohao9695@gmai.com, long.lan@nudt.edu.cn,
dreamkily@gmail.com

Abstract

Distinguishing key features in complex visual tasks is challenging. A novel approach treats image patches (tokens) as waves. By using both phase and amplitude, it captures richer semantics and specific invariances compared to pixel-based methods, and allows for feature fusion across regions for a holistic image representation. Based on this, we propose the Wave-wise Discriminative Transformer Tracker (WDT). During tracking, WDT represents features via phase-amplitude separation, enhancement, and mixture. First, we designed a Mutual Exclusive Phase-Amplitude Extractor (MEPAE) to separate phase and amplitude features with distinct semantics, representing spatial target info and background brightness respectively. Then, Wave-wise Feature Augmentation is carried out with two submodules: Phase-Amplitude Feature Augmentation and Mixture. The augmentation module disrupts the separated features in the same batch, and the mixture module recombines them to generate positive and negative waves. The original features are aggregated into the original wave. Positive waves have the same phase but different amplitudes, and negative waves have different phase components. Finally, self-supervised and tracking-supervised losses guide the global and local representation learning for original, positive, and negative waves, enhancing wave-level discrimination. Experiments on five benchmarks prove the effectiveness of our method.

1 Introduction

Visual Object Tracking (VOT) is pivotal in real - time computer vision, accurately tracking target objects within video sequences. At its core, tracking hinges on matching and localization, where feature extraction is a critical step determining a tracker's capacity to represent and distinguish targets. To ensure tracker robustness and accuracy, learning discriminative feature representations is essential. Tracking algorithms have significantly advanced, from handcrafted features to the current deep extraction methods. Using deep neural networks, these techniques learn effective features. However, research on deep discriminative tracking has bottlenecks. Prior pixel - based extraction limits information, so a new approach is urgently needed to enhance tracking performance.

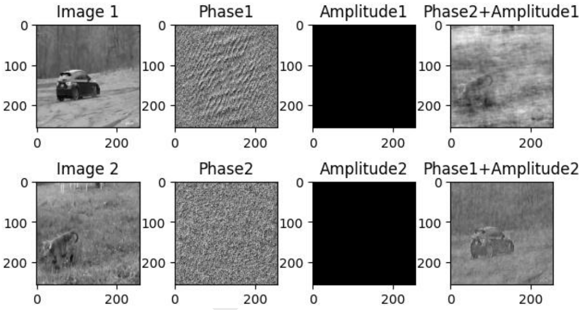

Figure 1: Schematic of the phase and amplitude characteristics of images, Image1 and Image2 from the GOT-10K test dataset.

Inspired by classical physics and quantum mechanics, wave-wise feature representation offers a new perspective: Images or image tokens are seen as a series of propagated wave-wise components, formed by fusing amplitude and phase via complex wave functions. This representation holds richer semantics, specific invariances, and strong information-aggregation capabilities, opening up new possibilities in visual tasks. It has been successfully applied in various scenarios. Early research relied on the Fourier transform to separately extract phase and amplitude features for specific tasks. But integrating this multi-step process into end-to-end deep learning was difficult. A recent work [Tang *et al.*, 2021] introduced a plug-and-play method in a Multi-Layer Perceptron architecture. Each token was treated as a visual waveform, and neural networks were used to model phase and amplitude extraction and aggregation across all tokens. Notably, phase mainly captures object-related spatial details like textures, edges, and outlines, while amplitude reflects global brightness, being more sensitive to background changes. To better use waveform features in VOT and distinguish targets from backgrounds, we aim to make the model extract phase and amplitude features distinctly from target and background semantic blocks. This ensures clear encoding of target structure and image brightness, resulting in more discriminative aggregated waveform features, rather than just repeating previous methods. In summary, waveform representation improves object tracking in three ways: 1) Capturing Semantic Details: Waveform representation extracts object shape and boundaries through phase shifts, and captures broader contrast changes through amplitude variations; 2) Global and Local Feature Fusion: Waveform features effectively combine information from various positions, facilitating a more comprehensive feature description for both individual images and image blocks. 3) Enhancing Robustness and Stability: The stability of waveform representation ensures improved tracking performance, particularly in challenging scenes.

Motivated by this, we develop a novel Wave-wise Discriminative Tracking method. For tracking tasks, the target's spatial structure and brightness information should be distinguished, with more emphasis on the former. For instance, in certain scenarios, lighting conditions might change, but the basic target structure remains relatively invariant. As shown in Figure 1, changes in image phase (phase swapping) lead to drastic variations, while amplitude swaps can still be distinguished by the human eye to identify objects.

Therefore, the core of our innovation lies in how to manipulate the phase and amplitude features within the framework of waveform representation to highlight the target structure and mitigate the background influence. First, Phase-Amplitude Feature Separation: Drawing on the findings of the literature [Tang *et al.*, 2021], we introduce an attention module to learn the weights of image tokens. Subsequently, a pair of mutually exclusive token weights are generated for phase and amplitude feature modeling respectively. The aim is to mainly model the phase information in the target region, while the amplitude information is mainly distributed in the background area. Integrating the above-mentioned processes constitutes the Mutual Exclusive Phase-Amplitude Extractor (MEPAE) module. Second, Phase-Amplitude Feature Enhancement: During model training, the separated phase-amplitude features within the same batch are randomly shuffled. This implies that each sample in the same batch contains both the original phase-amplitude features and the wave features from other samples. Third, Phase-Amplitude Feature Mixing: In this stage, the initial phase-amplitude features are incorporated into the original wave, and the enhanced phase-amplitude features are interwoven to generate multiple sets of positive and negative waves. Positive waves have the same phase but different amplitudes, and negative waves have different phases. Finally, Feature Discriminability Enhancement: To better enhance the discriminability of features, we combine self-supervised and tracking-supervised losses to constrain all waveform features from both global and local token perspectives. Specifically, the self-supervised loss is used to constrain the similarity between feature pairs such as the original-negative waves and positive-negative waves. The tracking-supervised loss constrains the token-level features of all waves. The above steps lead to the final waveform tracker, which can effectively extract feature information, improve the object tracking performance, and enhance the accuracy and robustness.

In summary, the main contributions of this work are threefold: **(1)** We introduce an innovative waveform-based feature representation, and conduct feature learning through the processes of phase-amplitude feature separation, augmentation, and mixture. **(2)** By combining self-supervised and supervised losses, we strengthen the overall and local learning of waveform features. The introduction of feature-level self-supervised learning into the tracking framework represents a novel plug-and-play integrated learning paradigm. **(3)** We construct a novel waveform discriminative tracking method based on the Transformer network, named Wave - wise Discriminative Tracker (WDT). This method has demonstrated excellent performance in multiple benchmark tests, including LaSOT, LaSOT$_{\text{ext}}$, TNL2K, UAV123, and GOT-10K.

2 Related Work

Transformer-based tracker. With the popularity of Transformer [Vaswani *et al.*, 2017] in computer vision, Transformer-based trackers have become a dominant category. SwinTrack [Lin *et al.*, 2022] and OSTrack [Ye *et al.*, 2022] ditch CNNs and fully embrace the transformer structure, achieving excellent results in multiple benchmarks. Recent studies have explored various aspects. [Cai *et al.*, 2024] uses the Transformer encoder to extract interaction features between the template and search region. [Xie *et al.*, 2024] employs the Transformer module for spatial image feature extraction and spatiotemporal video learning. [Zhao *et al.*, 2024] investigates the impact of different data augmentation on Transformer tracker performance. [Wei *et al.*, 2023] takes an autoregressive approach, using Transformer - encoded features to model and predict target trajectories. [Chen *et al.*, 2023] uses a ViT-based encoder for video frame feature extraction and a causal Transformer decoder for tracking results, simplifying the network and enhancing performance. [Li *et al.*, 2023] uses a Transformer backbone to extract joint features of the template and search image, facilitating interaction with text features for prediction.

Wave-wise features in Computer Vision. Phase and amplitude information are crucial features for images and have been successfully applied in several visual tasks [Wang *et al.*, 2025; Wang *et al.*, 2024b; Wang *et al.*, 2024c]. [Zhang *et al.*, 2023b] uses Semantic Frequency Prompt to interact with the feature map in the frequency domain, guiding the student model to learn valuable pixels. [Gu *et al.*, 2023] proposed AdaFuse, which employs the spatial-frequential fusion module on multi-scale features for adaptive multimodal image feature fusion in both frequency and spatial domains. [Cheng *et al.*, 2023] proposed the FGFL framework, generating frequency masks via an improved Grad-CAM algorithm in the frequency domain to filter key category information.

Self-supervised Learning. Self-supervised learning is one of the main types of machine learning, alongside unsupervised [Wang *et al.*, 2024d] and semi-supervised learning

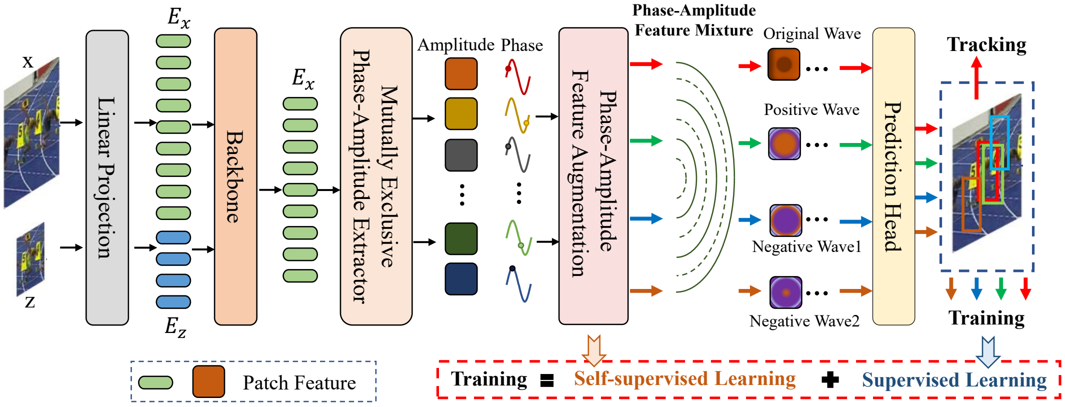
x
z
Linear Projection
E_x
E_z
Backbone
E_x
Mutually Exclusive Phase-Amplitude Extractor
Amplitude
Phase
Phase-Amplitude Feature Augmentation
Phase-Amplitude Feature Mixture
Original Wave
Positive Wave
Negative Wave1
Negative Wave2
Prediction Head
Tracking
Training
Patch Feature
Training = Self-supervised Learning + Supervised Learning

Figure 2: The pipeline of the Wave-wise Discriminative Tracker (WDT).

[Wang *et al.*, 2024a]. Self-supervised learning is highly effective in video tasks. For instance, [Qian *et al.*, 2024] trains a correlation network via object consistency in videos. [Ding *et al.*, 2025] improves object segmentation accuracy with spatiotemporal consistency. [Hamilton *et al.*, 2022] mines semantic consistency in feature maps for semantic segmentation.

3 Method

3.1 Framework of WDT

As the Figure 2 depicted, the framework of our WDT can be divided into five parts successively:

Backbone. We adopt the backbone of One-Stream One-Stage pipeline for image feature extraction and relation modeling. In this stage, the input template image z and search image x are split and flattened into a chain of patches N_z and N_x. And after a linear projection they will turn into a series of patch embeddings $\boldsymbol{E}_z \in \mathbb{R}^{N_z \times D}$ and $\boldsymbol{E}_x \in \mathbb{R}^{N_x \times D}$, where $D = P^2$ is the feature dimension of each token. Each of token will be added a position embedding and then the template embeddings $\boldsymbol{E}_z$ and search embeddings $\boldsymbol{E}_x$ will be concatenated as $E_{zx} = [E_z, E_x]$. And the E_{zx} will be processed by sequences of Transformer encoder layers. During this process, we finished the initial feature extraction, and at the same time, the relation between the search image and the exemplar image have been calculated.

Mutually Exclusive Phase-Amplitude Extractor (MEPAE). After the backbone network has fully learnt the relationship between the target and the search feature, it will output the transformed feature $\tilde{E}_{zx} \in R^{(N_z+N_x) \times P \times P}$. In the original framework, the corresponding search region $\tilde{E}_x \in R^{N_x \times P \times P}$ in $\tilde{E}_{zx}$ is fed into the **prediction head** to predict the tracking results. In between, MEPAE is embedded as a plug-and-play module. $\tilde{E}_x$ is taken as inputs and a new attention network is designed, which will acquire the labelled attention used to guide the phase and amplitude feature generation. Details will be presented in Sec. 3.2.

Phase-Amplitude Feature Augmentation (PAFA). To further enhance the learning of waveform features, we perform feature enhancement by shuffling the separated phase and amplitude features within a batch. This step is referred to as Phase-Amplitude Feature Augmentation (PAFA). We will discuss this in Section 3.3.

Phase-Amplitude Feature Mixture (PAFM). We insert a **PAFM** unit between the **PAFA** and the **prediction head**. In this process, the free combination of amplitude and phase terms will form a new set of feature samples, including the original waveform sample, the positive waveform samples with constant phase but only varying amplitude, and two groups of negative waveform samples with varying phase. Further we will impose supervised and self-supervised constraints on them to aid waveform feature learning. We will discuss this in later sections Sec. 3.3 and Sec. 3.4.

Prediction Head. We use a center prediction head for target classification and box regression. In addition, we added self-supervised loss and supervised loss for all waveform features. Both the Score map and the predicted box coordinates will be obtained after this structure.

In particular, features are extracted by MEPAE for both training and inference, except that for training there is feature augmentation and mixture, and all wave samples are output, whereas for inference only the predictions of the original waveform samples are output.

3.2 Mutually Exclusive Phase-Amplitude Extractor

Before the introduction of the structure of MEPAE, we present the representation of wave function first.

Wave-based representation. Denote one token feature output by the backbone as a wave $\tilde{\boldsymbol{t}}_j$:

$$\tilde{\boldsymbol{t}}_j = |\boldsymbol{t}_j| \odot e^{i\boldsymbol{\phi}_j}, j = 1, 2, \cdots, n \tag{1}$$

where $i^2 = -1$ and $|\cdot|$ is to calculate the absolute value and $\odot$ represents element-wise multiplication. $|\boldsymbol{t}_j|$ and the $\boldsymbol{\phi}_j$ are the amplitude and the phase we are concerned for. And the amplitude usually carries the content information of the token while the phase in periodic function $e^{i\boldsymbol{\phi}_j}$ represents the current state of the token in its period.

Network for Mutually Exclusive Tokens Attention. In order to better separate phase and amplitude and reflect different semantic properties, we introduce a simple attention

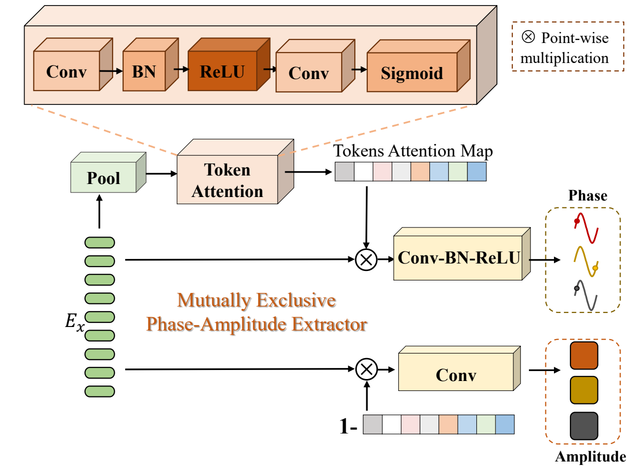

Figure 3: The detailed structure of the Mutually Exclusive Phase-Amplitude Extractor (MEPAE).

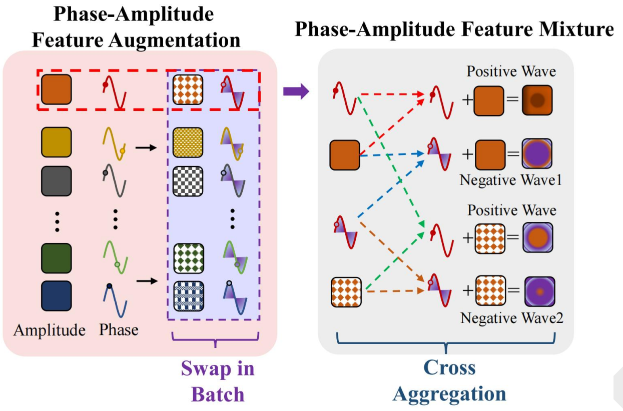

Figure 4: The process of wave-wise feature augmentation.

module for learning tokens weights. The tokens attention map M_p and its mutually exclusive attention $M_a = 1 - M_p$ are directly learned based on the output feature $\tilde{E}_x$, corresponding to phase and amplitude features modelling, respectively. Specifically, in two steps, first the initial features are pooled into $\tilde{E}_{px} \in R^{N_x \times 1 \times 1}$. And it is then fed into a token attention module for transformation learning consisting of multiple convolutional layers, Batch Normalization, ReLU and sigmoid functions, ultimately outputting a token attention map $M_p \in R^{N_x \times 1 \times 1}$.

Network for Phase-Amplitude Extraction. To extract the waveform features, phase and amplitude features need to be extracted separately first. In [Tang *et al.*, 2021], the neural network is innovatively used to express them separately. First we multiply the above attention maps M_p with the initial token features$\tilde{E}_x$ for phase extraction, and then go through a classical **Conv-BN-ReLU** module for feature transformation to obtain the final phase features $P = \{p_1, p_2, \cdots, p_{N_x}\}$. In another branch, we multiply $\tilde{E}_x$ by M_a for amplitude extraction, and then obtain the amplitude feature $A = \{a_1, a_2, \cdots, a_{N_x}\}$ through a **Conv** layer.

The above feature extraction process is jointly integrated into the mutually exclusive phase-amplitude extraction module, the structure of which is shown in Figure 3. The specific extraction process can be expressed as follows:

$$P = \{p_1, p_2, \cdots, p_{N_x}\} = f\left(M_p * \tilde{E}_x\right). \qquad (2)$$

$$A = \{a_1, a_2, \cdots, a_{N_x}\} = f\left(M_a * \tilde{E}_x\right). \qquad (3)$$

3.3 Wave-wise Feature Augmentation

To increase the diversity of waveform features, we designed a feature augmentation stage, which includes two steps: phase-amplitude feature augmentation and phase-amplitude feature mixture. In summary, in the prior step, we extracted phase and amplitude features separately using MEPAE. Subsequently, we separately expanded these features, before combining and rearranging them to produce a multitude of distinct waveform samples. The whole process is shown in Figure 4.

Phase-Amplitude Feature Augmentation. We expand phase and amplitude features by performing a **Swap in Batch** operation, as shown in Figure 4, which randomly disrupt the order of phase and amplitude features within a batch, resulting in new batch-sized groups of phase and amplitude features, respectively. Thus after this session, we already have two pairs of phase-amplitude features.

Phase-Amplitude Feature Mixture. Phase and amplitude features can be aggregated based on the wave function. By cross aggregating two pairs of phase-amplitude features two by two, we can obtain four samples, including the original wave $\tilde{t}^o$, a positive wave $\tilde{t}^p$ with the same phase as the original wave but different amplitudes, and two negative waves $\tilde{t}^{n_1}$ and $\tilde{t}^{n_2}$ with different phases. Note that in this process, each aggregation with phase and amplitude follows the mechanism of Eq. 4. According to Euler's formula, the process of wave aggregation is

$$\tilde{\boldsymbol{t}}_j = |\boldsymbol{t}_j| \odot \cos \boldsymbol{\theta}_j + i\,|\boldsymbol{t}_j| \odot \sin \boldsymbol{\theta}_j, j = 1, 2, \cdots, n \qquad (4)$$

By substituting the extracted phase and amplitude features separately and drawing on the properties of [Tang *et al.*, 2021], we can directly obtain the final fusion process:

$$\tilde{t}_j = \sum_k W^t_{jk} a_k \odot \cos p_k + W^{\mathrm{i}}_{jk} a_k \odot \sin p_k \qquad (5)$$

where W^t, W^i are both learnable weights. j and k denote the j-th row and k-th column of matrix W. a_k and p_k are amplitude and phase feature, respectively, and $\tilde{t}_j$ denotes the waveform feature of the j-th token.

3.4 Self-supervised Loss and Tracking-supervised Loss

To ensure the robustness and accuracy of the wive-wise features. We design two kinds of constraints for the above enhanced waves as shown in Figure 5. Note that the self-supervised loss deals with the overall features after all token waves are stretched, whereas the tracking supervised loss is computed with individual token waves as the unit of computation. Combining the two losses from the perspective of

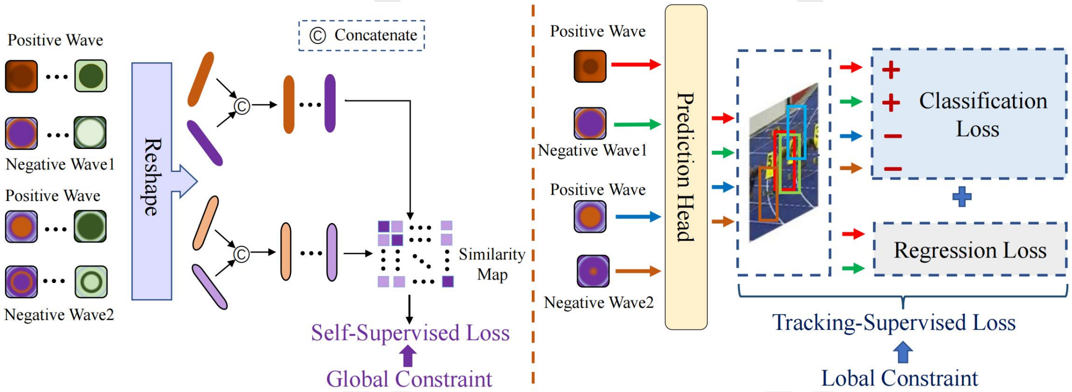

Figure 5: The process of self-supervised and supervised learning for wave samples.

global and local constraints can be more effective in improving feature discrimination.

Self-supervised loss. Observing the success of contrastive learning and relative loss fuctions, we adopt the similar self-supervised loss as [Radford *et al.*, 2021] for our feature augmentation. Specifically, we construct two new samples for learning self-supervised loss, i.e. $\tilde{\boldsymbol{T}}_r$ and $\tilde{\boldsymbol{T}}_s$, one sample is concatenated from the original wave $\tilde{t}^o$ and one negative wave $\tilde{t}^{n_1}$, and the other sample is concatenated from the positive wave $\tilde{t}^p$ and another negative wave $\tilde{t}^{n_2}$. Then we perform normalization on them, respectively:

$$\tilde{\boldsymbol{T}}_r = \mathcal{N}([\tilde{t}^o; \tilde{t}^{n_1}]) = \mathcal{N}(\tilde{\boldsymbol{T}}_r), \tag{6}$$

$$\tilde{\boldsymbol{T}}_s = \mathcal{N}([\tilde{t}^p; \tilde{t}^{n_2}]) = \mathcal{N}(\tilde{\boldsymbol{T}}_s), \tag{7}$$

where the $\mathcal{N}(\cdot)$ means the normalization operation.

We conduct tensor multiplication on the concatenated feature $\tilde{\boldsymbol{t}}$ and its transposed feature to compute a similarity logits.

$$sim = \tilde{T}_s \otimes \tilde{T}_r^{\ T} \tag{8}$$

where T means to calculate the transposed value.

Finally we perform a cross-entropy loss on the logits and groundtruth label:

$$\mathcal{L}_{\text{self}} = [\boldsymbol{CE}(sim, label) + \boldsymbol{CE}(sim^T, label)]/2, \tag{9}$$

where $\boldsymbol{CE}(\cdot)$ represents the CrossEntropy loss function.

Tracking-supervised loss. Following [Ye *et al.*, 2022], the supervised loss function for the original image features is:

$$\mathcal{L}_{\text{Sori}}\ (O_{ori}, g_t) = \mathcal{L}_{\text{cls}}\ + \lambda_{\text{iou}}\, \mathcal{L}_{\text{iou}}\ + \lambda_{L_1} \mathcal{L}_{L_1}. \tag{10}$$

where O_{ori} represents the output of WDT while it takes the original feature as input; g_t means the ground-truth label given by the dataset; $\mathcal{L}_{\text{cls}}$ is the weighted focal loss [Law and Deng, 2018], $\mathcal{L}_{\text{iou}}$ is the generalized IoU loss [Rezatofighi *et al.*, 2019] and $\mathcal{L}_{L_1}$ means the common l_1 loss.

Additionally, since in our method we generate a lot of positive samples with same phase as the original image. Then we put the same constraints on these features as below:

$$\mathcal{L}_{\text{Spos}}\ (O_{pos}, g_t) = \mathcal{L}_{\text{cls}}\ + \lambda_{\text{iou}}\, \mathcal{L}_{\text{iou}}\ + \lambda_{L_1} \mathcal{L}_{L_1}. \tag{11}$$

where the O_{pos} means the output of WDT while it takes the positive features as input.

And thus the complete supervised loss is:

$$\mathcal{L}_{\text{supervised}}\ = \mathcal{L}_{\text{Sori}}\ + \lambda_{\text{Spos}}\, \mathcal{L}_{\text{Spos}} \tag{12}$$

where the $\lambda_{\text{Spos}}\ = 1$ in our method.

To sum up, the total loss for our WDT is:

$$\mathcal{L}_{\text{total}}\ = \mathcal{L}_{\text{supervised}}\ + \lambda_{\text{self}} \mathcal{L}_{\text{self}} \tag{13}$$

where the $\lambda_{\text{self}} = 0.5$ in our method.

4 Experiments

4.1 Implementation

Model. Our tracker WDT is built upon ViT [Dosovitskiy *et al.*, 2020] and HiViT [Zhang *et al.*, 2023a], which adopts the template of 192×192 pixels and the search of 384×384 pixels, respectively abbreviated as WDT-ViT and WDT-HiViT. And all of our design modules are inserted after search tokens recovery and before prediction head.

Training. We implement our model in Python using PyTorch and train it with 8 NVIDIA A100 GPUs. And the test are conducted on a single NVIDIA RTX3070 GPU. For WDT-ViT, we set the batch size to 24, the weight decay to 10^{-4}, the learning rate for the backbone to 4×10^{-5} and the rest parameters to 4×10^{-4}, respectively. The learning rate decreases by a factor of 10 after 240 epochs. For WDT-HiViT, we set the batch size to 4, the initial learning rate of the backbone network to 2×10^{-5}, the learning rate of other parameters to 2×10^{-4}, and the weight decay to 10^{-4}. The total number of training epochs is 150, and the learning rate decreases by a factor of 10 after 120 epochs. The training datasets are COCO [Lin *et al.*, 2014], LaSOT [Fan *et al.*, 2018], GOT-10k [Huang *et al.*, 2018] and TrackingNet [Müller *et al.*, 2018]. The entire tracker is trained using both self-supervised loss and tracking-supervised loss, and the original waves, both the positive and the negatives, are tracked for prediction.

Inference. When tracking inference, we only output the tracking results of the **original wave** and test on five popular benchmarks: LaSOT [Fan *et al.*, 2018], LaSOT$_{\text{ext}}$ [Fan *et al.*, 2020], GOT-10k [Huang *et al.*, 2018] and UAV123 [Mueller *et al.*, 2016] and TNL2K [Wang *et al.*, 2021b].

Method	Source	LaSOT			$\text{LaSOT}_{\text{ext}}$			TNL2K	UAV123	GOT-10k		
		AUC	P_{Norm}	P	AUC	P_{Norm}	P	AUC	AUC	AO	$\text{SR}_{0.5}$	$\text{SR}_{0.75}$
SiamPRN++ [Li *et al.*, 2018]	CVPR19	49.6	56.9	49.1	34.0	41.6	39.6	41.3	61.3	51.7	61.6	32.5
DiMP [Bhat *et al.*, 2019]	ICCV19	56.9	65.0	56.7	39.2	47.6	45.1	44.7	65.4	61.1	71.7	49.2
Ocean [Zhang and Peng, 2020]	ECCV20	56.0	65.1	56.6	-	-	-	38.4	-	61.1	72.1	47.3
PACNet [Zhang *et al.*, 2021]	AAAI21	55.3	-	54.6	-	-	-	-	62.0	58.2	68.5	44.3
TransT [Chen *et al.*, 2021]	CVPR21	64.9	73.8	69.0	-	-	-	-	69.1	67.1	76.8	60.9
MixFormer [Cui *et al.*, 2022]	CVPR2022	70.1	79.9	76.3	-	-	-	-	70.4	71.2	80.0	67.8
SwinTrack-B [Lin *et al.*, 2022]	NeurIPS22	71.3	-	76.5	49.1	-	55.6	55.9	-	72.4	80.5	67.8
OSTrack [Ye *et al.*, 2022]	ECCV22	71.1	81.1	77.6	50.5	61.3	57.6	55.9	70.7	73.7	83.2	70.8
CiteTracker [Li *et al.*, 2023]	ICCV23	69.7	78.6	75.7	-	-	-	57.7	-	74.7	84.3	73.0
SeqTrack [Chen *et al.*, 2023]	CVPR23	71.5	81.1	77.8	50.5	61.6	57.5	56.4	68.6	74.5	84.3	71.4
ARTrack [Wei *et al.*, 2023]	CVPR23	72.6	81.7	79.1	51.9	62.0	58.5	**59.8**	70.5	75.5	84.3	74.3
DATr [Zhao *et al.*, 2024]	WACV24	71.0	80.7	77.5	51.8	62.7	59.0	-	69.7	74.2	84.1	71.1
STCFormer [Hu *et al.*, 2024]	AAAI24	71.5	81.5	78.0	52.0	63.0	59.6	57.7	70.8	74.3	84.2	72.6
AQAT [Xie *et al.*, 2024]	CVPR24	**72.7**	**82.9**	**80.2**	52.7	64.2	**60.8**	59.3	**71.2**	76.0	85.2	**74.9**
HIPTrack [Cai *et al.*, 2024]	CVPR24	**72.7**	**82.9**	79.5	**53.0**	**64.3**	60.6	-	70.5	**77.4**	**88.0**	74.5
WDT-ViT	Ours	71.4	81.3	77.9	52.3	63.2	59.9	57.5	**71.1**	74.5	83.8	71.8
WDT-HiViT	Ours	**73.0**	**83.3**	**80.7**	**53.3**	**64.8**	**61.2**	**59.7**	**71.2**	**76.1**	**85.4**	**75.3**

Table 1: Comparison with state-of-the-art trackers on five popular benchmarks. The best two results are shown in red and blue fonts.

4.2 Comparison with State-of-the-Arts

We compare our method with 15 state-of-the-art trackers, which include most representative methods in recent years. The comprehensive results are listed in Table 1.

LaSOT. LaSOT is a widely-used benchmark for long-term tracking. It contains 1,400 videos with more than 3.5M frames in total. And its evaluation metrics are the normalized precision (P_{Norm}), the precision (P) and the area under curve (AUC) of the success plot. Our trackers WDT-ViT and WDT-HiViT rank among the popular comparison methods. WDT-ViT is 0.3% higher than baseline OSTrack in AUC, while WDT-HiViT achieves 73.0% in AUC, 83.3% in P_{Norm} score and 80.7% in P score.

$\text{LaSOT}_{\text{ext}}$. $\text{LaSOT}_{\text{ext}}$ is an extension version of LaSOT. It involves 150 videos of 15 classes. The evaluation metrics are the same as the LaSOT. As an extended version of LaSOT, $\text{LaSOT}_{\text{ext}}$ contains 150 additional sequences of 15 object classes. Since it is released in recent two years, results on it are relatively fewer but we still conducted new research on it, with AUCs of 52.3% and 53.3%, which is superior to other comparison methods.

TNL2K. TNL2K is also a new dataset for natural language guided tracking. Its testing dataset contains 700 high diversity sequences. To improve the generality of tracking evaluation it introduces several adversarial samples and thermal images. Therefore TNL2K is a challenging benchmark currently. And we boost the performance in each metric and outperform other powerful counterparts like SwinTrack-B and CIA by a substantial margin. WDT-ViT achieves 57.5% in AUC, which surpass OSTrack by 1.6%. WDT-HiViT achieves 59.7% in AUC, and outperform other powerful counterparts like AQAT and STCFormer by a substantial margin.

UAV123. UAV123 is an important dataset from a low-altitude aerial perspective. We takes the AUC and the precision (P) to evaluate trackers' performance on it. On the UAV123, we are still performing well with 71.1% and 71.2% in AUC, compared to other trackers and are up significantly.

GOT-10k. GOT-10k is consist of 10k sequences for training and 180 videos for testing. It takes the average overlap (AO) and the success rate (SR) at overlap thresholds 0.5 and 0.75 as the evaluation metrics. We evaluated our WDT-ViT and WDT-HiViT on their official website, and the results showed that the improvements in AO, $\text{SR}_{0.5}$, and $\text{SR}_{0.75}$ were 0.8%, 0.6%, 1.0%, respectively, compared with OSTrack. WDT-HiViT obtains 76.1% in AO, 85.4% in $\text{SR}_{0.5}$ score and 75.3% in $\text{SR}_{0.75}$. Our tracker is also better compared to other trackers.

ViT/HiViT	PAE	MEPAE	WA-T	WA-S	LaSOT	LaSOT_{ext}	GOT-10K
ViT					71.1	50.5	73.7
ViT	✓				70.9	50.6	73.9
ViT		✓			71.2	50.9	74.1
ViT	✓		✓		71.1	51.2	74.0
ViT	✓		✓	✓	71.2	52.0	74.4
ViT		✓	✓		71.3	52.2	74.3
ViT		✓	✓	✓	71.4	52.3	74.5
HiViT					71.8	52.1	74.6
HiViT	✓				72.0	52.4	74.9
HiViT		✓			72.1	52.6	75.3
HiViT	✓		✓		72.4	52.7	75.1
HiViT	✓		✓	✓	72.8	52.9	75.8
HiViT		✓	✓		72.6	53.0	75.6
HiViT		✓	✓	✓	73.0	53.3	76.1

Table 2: Quantitative comparison results of WDT with different components.

4.3 Ablation Study

The contributions of Each Module. To judge the exact impact of each part, we test the models equipped with the different component on GOT-10K and LaSOT. When extracting phase and amplitude features, we introduce a mutually exclusive note of semantic separation of phase and amplitude features. As shown in Table 2, the PAE improves tracking performance based on the waveform representation, and the MEPAE improvement is more pronounced with the added mutually exclusive attention. To evaluate the impact of supervised loss and self-supervised loss, we tested the tracking performance of PAE and MEPAE separately with different

Tracking-Supervised Loss	Self-Supervised Loss	LaSOT	LaSOT$_{ext}$	GOT-10K
0.5	-	71.0	51.5	74.2
1	-	71.3	52.2	74.3
2	-	70.8	51.6	73.9
1	0.1	71.1	52.0	73.9
1	0.5	**71.4**	**52.3**	**74.5**
1	1	71.3	52.0	74.2

Table 3: Effect of different weights for ABS position loss and PN position loss of WDT-ViT. The best results are shown in bold font.

Tracker	Speed(fps)	MACs(G)	Params (M)
TrDiMP [Wang *et al.*, 2021a]	26	-	-
TransT [Chen *et al.*, 2021]	50	-	-
STARK-ST101 [Yan *et al.*, 2021]	32	18.5	42
SwinTrack-B [Lin *et al.*, 2022]	45	69.7	91
OSTrack [Ye *et al.*, 2022]	59.7	52.5	87
WDT-ViT	53.1	57.9	92

Table 4: Comparison on running speed and parameters with other representative Transformer-based trackers.

losses, where WA-T indicates that only tracking supervised loss is included and WA-S indicates that only self-supervised loss is included, As shown in Table 2, the supervised loss is essential for the whole training, and the tracking performance can be further improved with the addition of the self-supervised loss aid.

Parametric Analysis in Self-supervised Loss. We try different weights for λ for each loss we design. The range of testing weights of each loss is approximately determined by the scale of their values. According to Table 3, results show that 1 and 0.5 are best for Self-Supervised Loss and Tracking-Supervised Loss.

Speed and parameters. Our WDT tracker runs at a reduced speed compared to the baseline and its main computational effort is concentrated in the MEPAE module. The parameters of our WDT-ViT are 92M. As for multiply-accumulate computations (MACs), ours are 57.9G, about 5.4G larger than OSTrack. However, experiments show that there is not too much computational burden of our method. In the same operating environment, our tracker at 53.5 fps, which is slightly slower but still meets the real-time standard, shown in Table 4.

Complex environments. Figure 6 shows the accuracy of LaSOT in different attributes. We can see that our WDT is significantly better than AQAT in IV (illumination variation), FM (fast motion), FOC (full occlusion) , OV (out-of-view), and POC (partial occlusion) situations, demonstrating that WDT can maintain its advantages in more complex environments, especially scenes with significant dynamic changes.

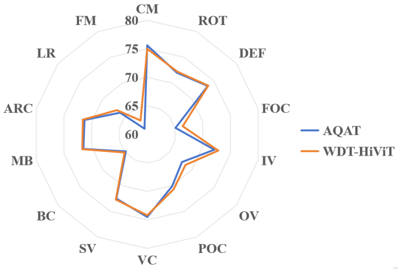

Figure 6: AUC scores of different attributes on LaSOT.

Visualization of tracking. In Figure 7 we display some representative results of our WDT. Obviously, WDT achieves quite accurate results in these scenarios. The results show that WDT can handle the deformation of the target well.

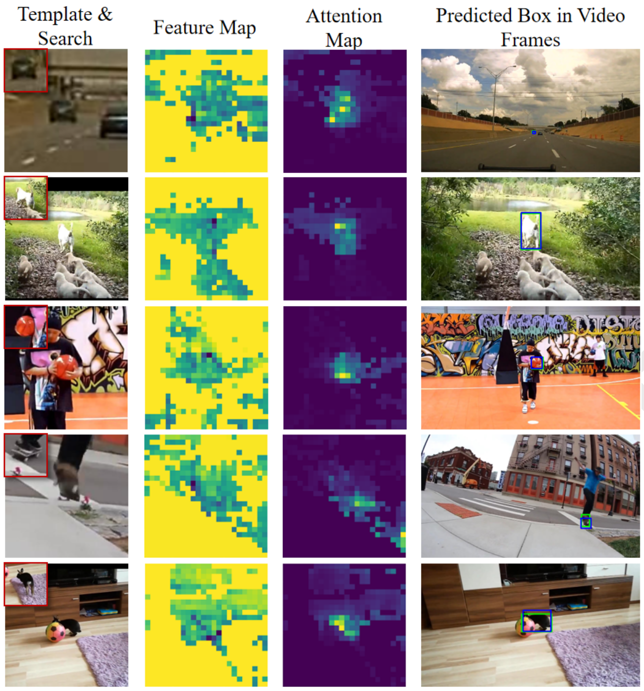

Figure 7: Visualization of the tracking results and feature maps. The first column shows the search images and the box results on it. The green rectangles are groundtruth boxes and the purple rectangles are predicted by our WDT. The template images are placed in the upper left corner. The second column shows the corresponding feature map. The third column shows attention map for corresponding search image. The fourth column shows the tracking results on corresponding frames.

5 Conclusion

In this paper, we propose a novel approach to feature learning in visual object tracking using wave representation. Treating each image patch as a wave function, we extract its phase and amplitude components separately. To enhance feature robustness, we perform augmentation by randomly recombining these components to form positive and negative pairs. A self-supervised constraint regulates this augmentation process, which is combined with tracking-supervised losses to guide wave feature learning at both global and local levels. By embedding these methods into ViT and HiViT, we develop trackers WDT-ViT and WDT-HiViT. Extensive experiments on benchmarks like LaSOT, TNL2K, UAV123, LaSOT$_{ext}$ and GOT-10K show their superior performance.

Acknowledgments

This work is supported by the Young Scientists Fund of the Hunan Natural Science Foundation (Grant No.2024JJ6474), the Youth Independent Innovation Science Fund Project of NUDT (Grant No.ZK24-08), the National Natural Science Foundation of China (Grant No.62406100), Tianjin Natural Science Foundation (Grant No.24JCQNJC00320) and Beijing Postdoctoral Research Foundation.

Contribution Statement

Huibin Tan and Mingyu Cao contribute equally and are regarded as co-first authors, noted by *. Long Lan and Mengzhu Wang contribute equally and are regarded as co-corresponding authors, noted by †.

References

[Bhat *et al.*, 2019] Goutam Bhat, Martin Danelljan, Luc Van Gool, and Radu Timofte. Learning discriminative model prediction for tracking. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6181–6190, 2019.

[Cai *et al.*, 2024] Wenrui Cai, Qingjie Liu, and Yunhong Wang. Hiptrack: Visual tracking with historical prompts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19258–19267, 2024.

[Chen *et al.*, 2021] Xin Chen, Bin Yan, Jiawen Zhu, Dong Wang, Xiaoyun Yang, and Huchuan Lu. Transformer tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8122–8131, 2021.

[Chen *et al.*, 2023] Xin Chen, Houwen Peng, Dong Wang, Huchuan Lu, and Han Hu. Seqtrack: Sequence to sequence learning for visual object tracking. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14572–14581, 2023.

[Cheng *et al.*, 2023] Hao Cheng, Siyuan Yang, Joey Tianyi Zhou, Lanqing Guo, and Bihan Wen. Frequency guidance matters in few-shot learning. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[Cui *et al.*, 2022] Yutao Cui, Jiang Cheng, Limin Wang, and Gangshan Wu. Mixformer: End-to-end tracking with iterative mixed attention. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13598–13608, 2022.

[Ding *et al.*, 2025] Shuangrui Ding, Rui Qian, Haohang Xu, Dahua Lin, and Hongkai Xiong. Betrayed by attention: A simple yet effective approach for self-supervised video object segmentation. In *European Conference on Computer Vision*, pages 215–233. Springer, 2025.

[Dosovitskiy *et al.*, 2020] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[Fan *et al.*, 2018] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. Lasot: A high-quality benchmark for large-scale single object tracking. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5369–5378, 2018.

[Fan *et al.*, 2020] Heng Fan, Hexin Bai, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Harshit, Mingzhen Huang, Juehuan Liu, Yong Xu, Chunyuan Liao, Lin Yuan, and Haibin Ling. Lasot: A high-quality large-scale single object tracking benchmark. *International Journal of Computer Vision*, 129:439–461, 2020.

[Gu *et al.*, 2023] Xianming Gu, Lihui Wang, Zeyu Deng, Ying Cao, Xingyu Huang, and Yue min Zhu. Adafuse: Adaptive medical image fusion based on spatial-frequential cross attention, 2023.

[Hamilton *et al.*, 2022] Mark Hamilton, Zhoutong Zhang, Bharath Hariharan, Noah Snavely, and William T Freeman. Unsupervised semantic segmentation by distilling feature correspondences. *arXiv preprint arXiv:2203.08414*, 2022.

[Hu *et al.*, 2024] Kun Hu, Wenjing Yang, Wanrong Huang, Xianchen Zhou, Mingyu Cao, Jing Ren, and Huibin Tan. Sequential fusion based multi-granularity consistency for space-time transformer tracking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 12519–12527, 2024.

[Huang *et al.*, 2018] Lianghua Huang, Xin Zhao, and Kaiqi Huang. Got-10k: A large high-diversity benchmark for generic object tracking in the wild. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:1562–1577, 2018.

[Law and Deng, 2018] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. *International Journal of Computer Vision*, 128:642–656, 2018.

[Li *et al.*, 2018] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4277–4286, 2018.

[Li *et al.*, 2023] Xin Li, Yuqing Huang, Zhenyu He, Yaowei Wang, Huchuan Lu, and Ming-Hsuan Yang. Citetracker: Correlating image and text for visual tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9974–9983, 2023.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

[Lin *et al.*, 2022] Liting Lin, Heng Fan, Zhipeng Zhang, Yong Xu, and Haibin Ling. Swintrack: A simple and strong baseline for transformer tracking. *Advances in Neural Information Processing Systems*, 35:16743–16754, 2022.

[Mueller *et al.*, 2016] Matthias Mueller, Neil G. Smith, and Bernard Ghanem. A benchmark and simulator for uav tracking. In *European Conference on Computer Vision*, 2016.

[Müller *et al.*, 2018] Matthias Müller, Adel Bibi, Silvio Giancola, Salman Al-Subaihi, and Bernard Ghanem. Trackingnet: A large-scale dataset and benchmark for object tracking in the wild. *ArXiv*, abs/1803.10794, 2018.

[Qian *et al.*, 2024] Zekun Qian, Ruize Han, Junhui Hou, Linqi Song, and Wei Feng. Vovtrack: Exploring the potentiality in videos for open-vocabulary object tracking. *arXiv preprint arXiv:2410.08529*, 2024.

[Radford *et al.*, 2021] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

[Rezatofighi *et al.*, 2019] Seyed Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666, 2019.

[Tang *et al.*, 2021] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10925–10934, 2021.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *ArXiv*, abs/1706.03762, 2017.

[Wang *et al.*, 2021a] Ning Wang, Wen gang Zhou, Jie Wang, and Houqiang Li. Transformer meets tracker: Exploiting temporal context for robust visual tracking. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1571–1580, 2021.

[Wang *et al.*, 2021b] Xiao Wang, Xiujun Shu, Zhipeng Zhang, Bo Jiang, Yaowei Wang, Yonghong Tian, and Feng Wu. Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13758–13768, 2021.

[Wang *et al.*, 2024a] Mengzhu Wang, Jiao Li, Houcheng Su, Nan Yin, Liang Yang, and Shen Li. Graphcl: Graph-based clustering for semi-supervised medical image segmentation. *arXiv preprint arXiv:2411.13147*, 2024.

[Wang *et al.*, 2024b] Mengzhu Wang, Junze Liu, Ge Luo, Shanshan Wang, Wei Wang, Long Lan, Ye Wang, and Feiping Nie. Smooth-guided implicit data augmentation for domain generalization. *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[Wang *et al.*, 2024c] Mengzhu Wang, Yuehua Liu, Jianlong Yuan, Shanshan Wang, Zhibin Wang, and Wei Wang. Inter-class and inter-domain semantic augmentation for domain generalization. *IEEE Transactions on Image Processing*, 33:1338–1347, 2024.

[Wang *et al.*, 2024d] Mengzhu Wang, Shanshan Wang, Xun Yang, Jianlong Yuan, and Wenju Zhang. Equity in unsupervised domain adaptation by nuclear norm maximization. *IEEE Transactions on Circuits and Systems for Video Technology*, 34(7):5533–5545, 2024.

[Wang *et al.*, 2025] Mengzhu Wang, Houcheng Su, Sijia Wang, Shanshan Wang, Nan Yin, Li Shen, Long Lan, Liang Yang, and Xiaochun Cao. Graph convolutional mixture-of-experts learner network for long-tailed domain generalization. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[Wei *et al.*, 2023] Xing Wei, Yifan Bai, Yongchao Zheng, Dahu Shi, and Yihong Gong. Autoregressive visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9697–9706, 2023.

[Xie *et al.*, 2024] Jinxia Xie, Bineng Zhong, Zhiyi Mo, Shengping Zhang, Liangtao Shi, Shuxiang Song, and Rongrong Ji. Autoregressive queries for adaptive tracking with spatio-temporal transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19300–19309, 2024.

[Yan *et al.*, 2021] Bin Yan, Houwen Peng, Jianlong Fu, Dong Wang, and Huchuan Lu. Learning spatio-temporal transformer for visual tracking. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10428–10437, 2021.

[Ye *et al.*, 2022] Botao Ye, Hong Chang, Bingpeng Ma, and S. Shan. Joint feature learning and relation modeling for tracking: A one-stream framework. *ArXiv*, abs/2203.11991, 2022.

[Zhang and Peng, 2020] Zhipeng Zhang and Houwen Peng. Ocean: Object-aware anchor-free tracking. *ArXiv*, abs/2006.10721, 2020.

[Zhang *et al.*, 2021] Dawei Zhang, Zhonglong Zheng, Riheng Jia, and Minglu Li. Visual tracking via hierarchical deep reinforcement learning. In *AAAI Conference on Artificial Intelligence*, 2021.

[Zhang *et al.*, 2023a] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *The Eleventh International Conference on Learning Representations*, 2023.

[Zhang *et al.*, 2023b] Yuan Zhang, Tao Huang, Jiaming Liu, Tao Jiang, Kuan Cheng, and Shanghang Zhang. Freekd: Knowledge distillation via semantic frequency prompt. *IEEE*, 2023.

[Zhao *et al.*, 2024] Jie Zhao, Johan Edstedt, Michael Felsberg, Dong Wang, and Huchuan Lu. Leveraging the power of data augmentation for transformer-based tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6469–6478, 2024.