drones

MDPI

Article

CMDN: Pre-Trained Visual Representations Boost Adversarial Robustness for UAV Tracking

Ruilong Yu, Zhewei Wu, Qihe Liu *, Shijie Zhou, Min Gou and Bingchen Xiang

School of Information and Software Engineering, University of Electronic Science and Technology of China, No. 4, Section 2 Jianshebei Road, Chengdu 610000, China; 202221090211@std.uestc.edu.cn (R.Y.); zheweiwu@std.uestc.edu.cn (Z.W.); sjzhou@uestc.edu.cn (S.Z.); mingou@std.uestc.edu.cn (M.G.); 202321090216@std.uestc.edu.cn (B.X.)

* Correspondence: qiheliu@uestc.edu.cn

Abstract: Visual object tracking is widely adopted to unmanned aerial vehicle (UAV)-related applications, which demand reliable tracking precision and real-time performance. However, UAV trackers are highly susceptible to adversarial attacks, while research on developing effective adversarial defense methods for UAV tracking remains limited. To tackle these challenges, we propose CMDN, a novel pre-processing defense network that effectively purifies adversarial perturbations by reconstructing video frames. This network learns robust visual representations from video frames, guided by meaningful features from both the search region and the template. Comprehensive experiments on three benchmarks demonstrate that CMDN is capable of enhancing a UAV tracker's adversarial robustness in both adaptive and non-adaptive attack scenarios. In addition, CMDN maintains stable defense effectiveness when transferred to heterogeneous trackers. Real-world tests on the UAV platform also validate its reliable defense effectiveness and real-time performance, with CMDN achieving 27 FPS on NVIDIA Jetson Orin 16 GB (25 W mode).

Keywords: unmanned aerial vehicle; adversarial defense; visual object tracking

check for updates

Citation: Yu, R.; Wu, Z.; Liu, Q.; Zhou, S.; Gou, M.; Xiang, B. CMDN: Pre-Trained Visual Representations Boost Adversarial Robustness for UAV Tracking. *Drones* **2024**, *8*, 607. https://doi.org/10.3390/drones8110607

Academic Editor: Xiwang Dong

Received: 25 September 2024
Revised: 14 October 2024
Accepted: 22 October 2024
Published: 23 October 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Unmanned aerial vehicle (UAV) tracking has attracted growing attention recently for various applications [1–3], aiming to estimate the position or trajectory of a specific target from an aerial perspective. With the development of deep neural networks (DNNs), DNN-based UAV trackers have made significant progress in solving this dilemma. Unlike ordinary visual object tracking (VOT), UAV tracking faces challenges from difficult scenarios (motion blur, partial occlusions, lighting variations, etc.) and constrained computational resource, raising two main concerns: tracking precision and real-time performance. Currently, the most popular approaches for UAV tracking are based on Siamese neural networks (SNNs), such as SiamRPN++ [4], SiamAPN [5], SiamAPN++ [6], and HiFT [7], due to their excellent tracking efficiency and precision.

However, due to the vulnerability of DNNs to adversarial examples [8,9], a UAV tracker may be misled by minor perturbations in the input images, leading to incorrect model decisions. Based on the manner in which perturbations are generated, existing adversarial attack methods against VOT can be roughly categorized into two types: (1) The first type is optimization-based attacks (such as SPARK [10], RTAA [11], IoU Attack [12], One-Shot Attack [13]) that iteratively optimize perturbations using gradient information queried from a tracker. These attacks could cause significant degradation in tracking precision but introduce a large amount of computational time. (2) The second type is DNN-based attacks (such as CSA [14], DFA [15], Ad^{2}Attack [16]) that train perturbation generators offline with well-designed attack loss functions, which could operate efficiently to generate perturbations with stable attack effectiveness. The presence of these adversarial

attack methods poses a serious threat to UAV tracking. Thus, it is critically necessary to develop an effective and efficient adversarial defense approach suitable for UAV tracking.

Currently, the exploration of adversarial defense strategies for VOT is quite insufficient. While AADN [17] has remarkable processing speed, it struggles to transfer its effectiveness to transformer-based trackers and faces challenges in adaptive attack scenarios, whereas LRR's [18] defense network introduces excessive time consumption to the tracker, despite of its excellent defense effectiveness. Therefore, existing defense methods are not suitable for UAV tracking tasks that require stable tracking precision and real-time performance. Researching adversarial defense methods feasible for UAV trackers is an urgently needed issue.

To address these challenges, we propose a novel adversarial defense network called CMDN for UAV tracking. Specifically, inspired by Masked Image Modeling [19–21], a pair of complementary binary masks is generated randomly to mask each half of the video frame. With the guidance of pre-trained visual representations learned from search regions and a template, the defense network is capable of reconstructing the masked areas and purifying potential adversarial perturbations simultaneously, as illustrated in Figure 1. Essentially, CMDN is placed in front of the tracker backbone as a input preprocessor and can be smoothly integrated with UAV trackers in a plug-and-play manner.

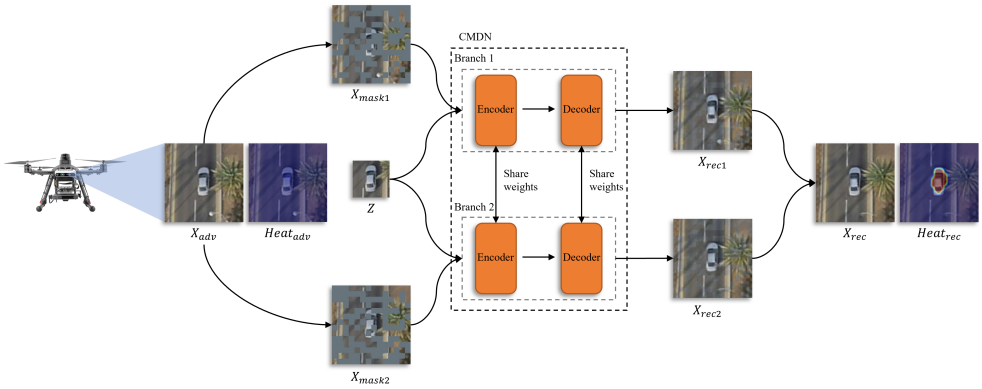


Figure 1. The visualized inference pipeline of CMDN. The adversarial search region is masked by two precisely complementary binary masks and is reconstructed through the combined output of two weight-sharing network branches. As shown in $Heat_{adv}$ and $Heat_{rec}$, CMDN effectively recovers the tracker's response to the tracking target. Better viewed in colors and textures with zoom-in.

Extensive experiments on the UAV123 [22], OTB100 [23], and VOT2018 [24] benchmarks demonstrate CMDN's remarkable robustness against adversarial attacks in both adaptive and non-adaptive scenarios. It is able to effectively purify potential adversarial perturbations while maintaining excellent tracking precision on clean video frames. Figure 2 illustrates that CMDN maintains stable defense effectiveness when facing targets of different categories, and even restores target loss. In addition, experiments on heterogeneous trackers without retraining the defense network demonstrate CMDN's outstanding transferability across UAV trackers based on SNN and Vision Transformer (ViT) [25]. In terms of real-world testing, our defense network attains 27 FPS on a UAV platform while maintaining stable effectiveness, which proves its practicality and stability.

In summary, this work makes the following contributions:

1. We propose CMDN, a novel adversarial defense approach tailored for UAV tracking. CMDN demonstrates remarkable efficiency and effectiveness in defense, aligning well with the high-precision and real-time demands of UAV tracking.
2. Experiments conducted in three widely used benchmarks illustrate that our defense network can considerably strengthen the robustness of an SNN-based tracker against adversarial attacks in both adaptive and non-adaptive scenarios. CMDN also demonstrates outstanding transferability on heterogeneous trackers, which makes it convenient to deploy CMDN in a plug-and-play manner.

3. The real-world testing conducted on a UAV platform verifies the efficiency and effectiveness of our defense network on edge devices.

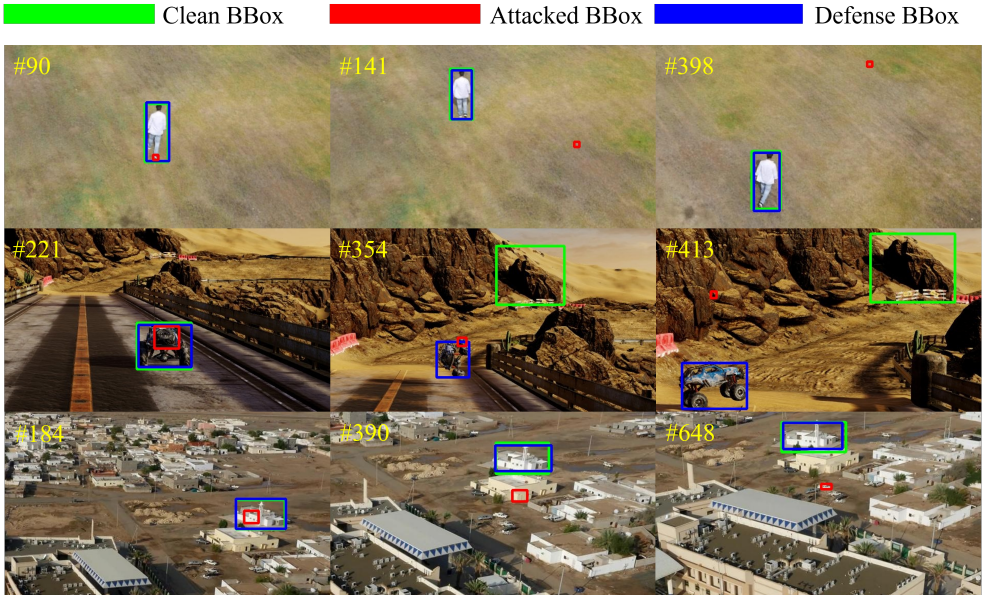


Figure 2. The visualized tracking results of defense effectiveness against CSA on three sequences (person, car, and building) from UAV123. The yellow numbers represent the indices of video frames. Better viewed in color with zoom-in.

Specifically, Section 2 introduces the background knowledge related to this work. Section 3 describes the loss function and reconstruction process of CMDN, and Section 4 details the implementation with the quantitative results of experiments. Section 5 discusses the contributions and potential weaknesses of CMDN, and explores their future solutions.

2. Related Works

2.1. UAV Tracking

UAV tracking has gained considerable attention in various applications [1–3]. Existing UAV trackers can be broadly categorized into methods that include SNN-based trackers and ViT-based trackers.

SNN-based trackers have become the mainstream in UAV tracking owing to their good balance between precision and efficiency. SiamFC [26] first proposed a naive cross-correlation strategy using a fully Convolutional Neural Network to learn feature similarity functions. SiamRPN++ [4] introduced deep cross-correlation in channel dimension and spatial-aware sampling strategies, which considerably enhance the performance of an SNN-based tracker. To tackle the issue of limited computing resources in UAV tracking, SiamAPN [5] designed an anchor proposal network for lightweight anchor generation. SiamAPN++ [6] introduced an attentional aggregation network for raising the representation ability of semantic features, which further improves the precision against various special challenges in UAV tracking. HiFT [7] proposed a slim hierarchical feature transformer for effective and efficient multi-level feature fusion.

ViTs [25] are designed as alternatives to traditional Convolutional Neural Networks (CNNs) [27,28] in computer vision. They adopt the transformer [29] architecture that was originally developed for Natural Language Processing (NLP) tasks. ViTs process images by dividing them into fixed-size patches, which are then flattened into sequences similar to how words are embedded in NLP tasks. These sequences are fed into the transformer model to learn hierarchical feature representations. Recently, ViTs have demonstrated superior performance in image classification compared with CNNs [30,31]. Moreover, with the development of lightweight ViTs [32,33], the major concern regarding inference speed on resource-constrained edge devices has been significantly alleviated, leading to several

attempts in UAV tracking. Aba-ViTrack [34] integrates feature learning and template-search coupling into an efficient one-stream ViT instead of an extra heavy relation module, and employs an adaptive and background-aware token computation method to reduce inference time consumption. SiamSTM [35] utilizes the lightweight transformer to encode robust target appearance features while using the multiple matching networks to fully perceive response map information and enhance the tracker's ability to distinguish between the target and background.

Although these various kinds of UAV trackers offer outstanding tracking precision and efficiency, they remain vulnerable to adversarial attacks [10–16], thereby posing a serious threat to the robustness of UAV tracking systems.

2.2. Adversarial Attack and Defense in Visual Object Tracking

Over the past few years, researchers have developed a number of adversarial attacks against VOT. SPARK [10] proposes an online incremental attack method that utilizes information from past frames. IoU Attack [12] iteratively adjusts the direction of perturbation according to the predicted IoU scores of bounding boxes and transfers the preceding perturbation into the next frames. One-shot Attack [13] leverages dual attention loss and background distraction loss to add slight perturbations merely into a tracking template to blind SNN-based trackers. CSA [14] trains an effective and efficient perturbation generator via cooling loss to suppress the hot region and shrinking loss to shrink the predicted bounding box. Ad^{2}Attack [16] first uses direct downsampling and super-resolution technology to imperceptibly lose pixel information. Then a residual spatial enhancement module is proposed to targetedly express the image feature, and an attack loss function is designed to drift the predicted bounding box. The perturbation generator is quite slim, especially for UAV tracking. These attacks could result in a significant decrease in tracking precision, leading to substantial financial loss and raising critical security concerns in the UAV scenarios. Consequently, the exploration of defense strategies to boost the adversarial robustness of UAV tracking is of paramount importance.

Unfortunately, only a limited number of researchers have conducted preliminary exploration in the field of adversarial defense for VOT. AADN [17] employs adversarial training to develop an auxiliary defense network, specifically utilizing a dual-loss function to generate adversarial examples that target both the classification and regression branches of the tracker simultaneously. LRR [18] utilizes semantic text guidance extracted from a language-image model, such as CLIP, to build a spatial–temporal implicit representation. This approach reconstructs incoming frames, maintaining consistency in both the semantic and visual aspects with the object of interest and its clean counterparts. Although these two methods achieve notable defense effectiveness, they still exhibit flaws. Specifically, AADN's effectiveness is challenging to transfer to ViT-based trackers and confronts difficulties when dealing with adaptive attack scenarios. LRR has a complex architecture that leads to significant computational overhead, making it not well suited for UAV tracking, which requires good real-time performance.

In this work, a novel adversarial defense network, designated as CMDN, which is specifically tailored for UAV tracking, is proposed to tackle this predicament. By integrating a complementary reconstruction process with pre-trained visual representations, CMDN demonstrates remarkable defense effectiveness in both adaptive scenarios and non-adaptive scenarios. In addition, when implemented on a UAV platform with SiamAPN [5], CMDN achieves 27 FPS, fulfilling the real-time requirement of UAV tracking.

2.3. Masked Image Modeling

In recent years, self-supervised learning has garnered increasing attention in machine learning and computer vision. This approach enables models to learn a rich feature representation, thereby outperforming the effectiveness of supervised alternatives. As the counterpart of masked language modeling (MLM) [36,37] in pretraining large-scale language models, masked image modeling (MIM) [19–21] has emerged as the prevailing

paradigm in a self-supervised pretraining vision model with the breakthrough of ViT. MIM is based on the idea that mask patches of the input image reconstruct the missing pixels. Concretely, the encoder is trained to extract useful features only from the visible patches, and the decoder learns to reconstruct the missing part from the latent representation and mask tokens.

Consequently, this technique of masking images, followed by their restoration, has inspired the development of our defense network. By achieving high-quality reconstruction of video frames, CMDN is capable of purifying the potential perturbation while maintaining excellent tracking precision on clean video frames.

3. Methodology

3.1. Problem Definition

To begin with, we briefly review the pretraining process of MAE [20] as an example of MIM. The framework includes an encoder f_θ and a decoder g_ϕ, where θ and ϕ denote their network parameters. The original input image $I_{ori} \in \mathbb{R}^{H\times W\times C}$ is first split into non-overlap patches, where H, W, and C denote height, width, and channel dimensions, respectively. Then, a binary mask M_α is randomly sampled at a masking ratio α. Therefore, the remaining visible patches I_{vis} are given by Equation (1), where $\odot$ denotes the element-wise product.

$$I_{vis} = I_{ori} \odot (1 - M_\alpha) \tag{1}$$

After that, I_{vis} are passed through f_θ with added positional embeddings. Subsequently, the latent vectors I_{latent} corresponding to encoded visible tokens and learnable mask tokens are fed into g_ϕ. Positional embeddings are added to all tokens to provide location information of the mask tokens in the original image. g_ϕ outputs the reconstructed patches I_{rec}. The reconstruction loss is calculated by the mean squared error (MSE) between the original patches I_{ori} and the reconstructed patches I_{rec}. The whole pretraining pipeline of MAE is illustrated in Figure 3.

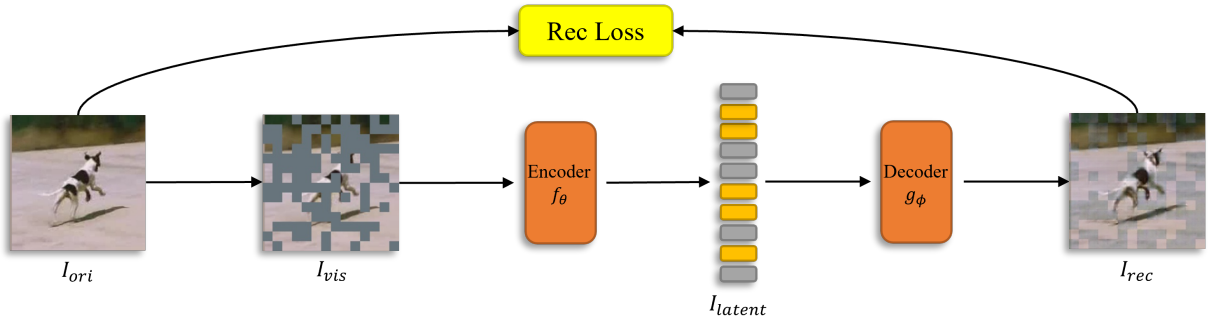


Figure 3. The pretraining framework of MAE. The grey and golden blocks represent masked tokens and visible tokens, respectively.

Next, we introduce the framework of SNN-based trackers. Given a template Z, the tracker is supposed to predict the location and shape of the target in the subsequent frames via the search region X. Concretely, the backbone extracts the features maps F_Z and F_X from Z and X. With the similarity map calculated between F_Z and F_X, the classification branch computes the confidence of candidate boxes to categorize the target and background, generating the classification map, while the regression branch adjusts the location and the shape of candidate boxes, generating the regression map. Finally, the tracker outputs the predicted bounding box after ranking all the candidate boxes based on their confidence scores. The rough framework of an SNN-based tracker is shown in Figure 4.

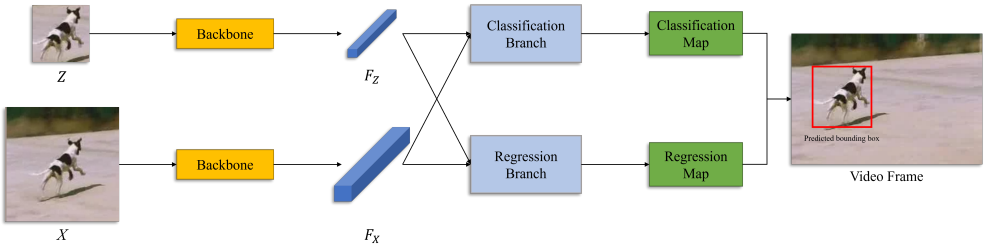


Figure 4. The framework of an SNN-based tracker.

3.2. Reconstruction Loss Function

We design a novel reconstruction loss $\mathcal{L}_{rec}$ to reconstruct images' appearance while retaining visual features crucial for object tracking, thereby enhancing both tracking precision and adversarial robustness. It consists of pixel loss $\mathcal{L}_{pix}$ and feature loss $\mathcal{L}_{feat}$, formulated as Equation (2), where ξ_1 and ξ_2 control the weight of two components.

$$\mathcal{L}_{rec} = \xi_1 \mathcal{L}_{pix} + \xi_2 \mathcal{L}_{feat} \tag{2}$$

As indicated in Equation (3), the mean squared error (MSE) between the original search region X_{ori} and the reconstructed search region X_{fin} is employed as $\mathcal{L}_{pix}$ to reduce their distance in the pixel space. In addition, $\mathcal{L}_{pix}$ is averaged over the number num of the masked patches.

$$\mathcal{L}_{pix} = \frac{\|X_{ori}, X_{fin}\|_2}{num} \tag{3}$$

Guided by $\mathcal{L}_{pix}$, the defense network is capable of reconstructing the appearance of a tracking target, thus maintaining the tracking precision on clean examples. However, when confronted with adversarial attacks, simply predicting pixel values from masked images may introduce adversarial perturbations into the reconstructed video frames, which could still lead to tracking failure. Therefore, we introduce $\mathcal{L}_{feat}$ to learn high-level representations in feature space, preserving critical information for similarity matching. Specifically, the original search region X_{ori} and the final reconstructed search region X_{fin} are input separately into the tracker's backbone T to extract the feature maps F_{ori} and F_{fin}. $\mathcal{L}_{feat}$ is then calculated as the MSE between these two feature maps as follows:

$$\begin{aligned} F_{ori} &= T(X_{ori}) \\ F_{fin} &= T(X_{fin}) \\ \mathcal{L}_{feat} &= \|F_{ori}, F_{fin}\|_2 \end{aligned} \tag{4}$$

3.3. Complementary Reconstruction Process

The MAE pretraining framework in Section 3.1 reconstructs only a portion of the image. If under adversarial attacks, the remaining original parts of the image may still contain unpurified adversarial perturbations. Therefore, it is essential to reconstruct the entire image rather than a random part. Building upon the work in Section 3.2, we propose an effective reconstruction approach with complementary masks to achieve this objective.

Given by Equation (5), a binary mask M_1 is randomly sampled at a masking ratio of 50%, and then inverted to form another mask M_2. These two masks are applied to the original search region X_{ori} to generate X_{vis1} and X_{vis2}, which are precisely complementary.

$$\begin{aligned} X_{vis1} &= X_{ori} \odot (1 - M_1) \\ X_{vis2} &= X_{ori} \odot (1 - M_2) \end{aligned} \tag{5}$$

Inspired by jointly masked encoding [38], we add another parallel template branch to improve the quality of reconstruction. We use the encoder f_θ to jointly encode Z, X_{vis1} and X_{vis2} to obtain the latent vectors Z_{latent}, $X_{latent1}$, and $X_{latent2}$, formulated as Equation (6).

We anticipate that this approach could enable the encoder to capture redundant appearance representations within the template, which encompasses prior knowledge of the target object, thereby guiding the defense network to reconstruct more effectively.

$$Z_{latent}, X_{latent1}, X_{latent2} = f_\theta(Z, X_{vis1}, X_{vis2}) \tag{6}$$

As the template is not reconstructed in the training stage, only $X_{latent1}$ and $X_{latent2}$ are passed through the decoder g_ϕ to produce X_{rec1} and X_{rec2}. Finally, utilizing the positional information from M_1 and M_2, we concatenate the reconstructed parts from X_{rec1} and X_{rec2} to obtain the fully reconstructed search region X_{fin}. This process can be written as Equation (7). Figure 5 illustrates the whole training pipeline of CMDN.

$$\begin{aligned} X_{rec1} &= g_\phi(X_{latent1}) \\ X_{rec2} &= g_\phi(X_{latent2}) \\ X_{fin} &= X_{rec1} \odot M_1 + X_{rec2} \odot M_2 \end{aligned} \tag{7}$$

In summary, CMDN reconstructs the entire search region, utilizing valuable features extracted from both the tracking template and the search region, which ensures that the tracker maintains high precision on clean examples and the majority of potential adversarial perturbations are purified. Figure 6 presents the visualized search regions with their corresponding heatmaps in clean, adversarial, and defense scenarios, which prove that CMDN successfully recovers the tracker's response across different target categories (person, car, boat, and building) when under adversarial attack. The detailed training process of CMDN is demonstrated in Algorithm 1.

Algorithm 1 Framework of pretraining process of proposed CMDN

Input: training set D, training epochs E, tracker backbone T
Output: trained defense network parameters θ for encoder, ϕ for decoder

for i in range [1, E] **do**
  for random training batch $\{Z, X\} \in D$ **do**
    Split template Z and search region X into non-overlap patches.
    Randomly sample a binary mask M_1 at a masking ratio of 50%.
    Invert M_1 to M_2.
    $X_{vis1} = X_{ori} \odot (1 - M_1), X_{vis2} = X_{ori} \odot (1 - M_2)$
    $Z_{latent}, X_{latent1}, X_{latent2} = f_\theta(Z, X_{vis1}, X_{vis2})$
    $X_{rec1} = g_\phi(X_{latent1}), X_{rec2} = g_\phi(X_{latent2})$
    $X_{fin} = X_{rec1} \odot M_1 + X_{rec2} \odot M_2$
    $F_{ori} = T(X_{ori}), F_{fin} = T(X_{fin})$
    Calculate the pixel loss $\mathcal{L}_{pix}$ using Equation (3).
    Calculate the feature loss $\mathcal{L}_{feat}$ according to Equation (4).
    Compute the gradient of $\mathcal{L}_{rec}$ to defense network parameters θ and ϕ and update them with the AdamW optimizer.
  end for
end for
return trained defense network parameters θ and ϕ

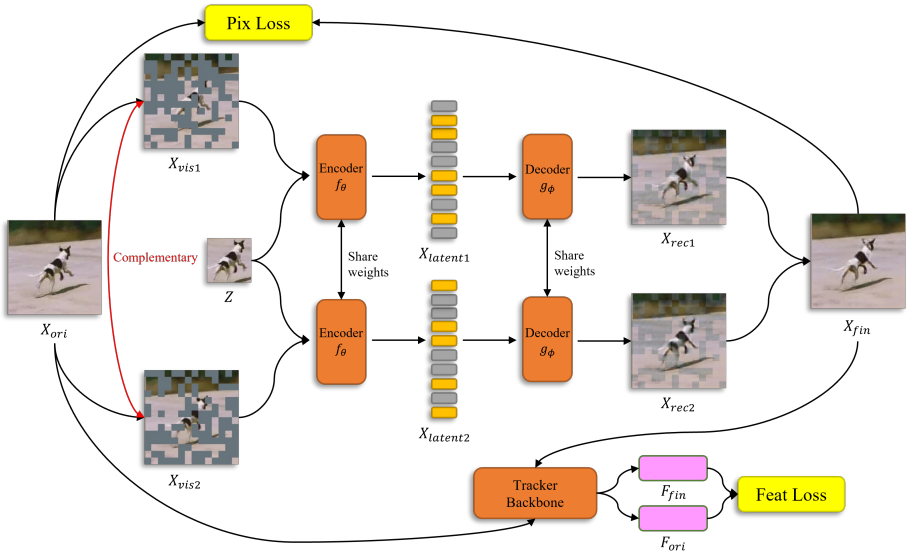


Figure 5. The training pipeline of CMDN. The grey and golden blocks represent masked tokens and visible tokens, respectively.



Figure 6. The visualized search regions with corresponding heatmaps from the UAV123 dataset. Columns from left to right show the original search regions, search regions with clean heatmaps, search regions with adversarial heatmaps attacked by CSA, and search regions with defense heatmaps.

4. Experiments

In this section, we comprehensively describe the details of our experiments. In Sections 4.3 and 4.4, we integrate CMDN with SiamRPN++ to evaluate its robustness

against adversarial attacks in both non-adaptive and adaptive attack scenarios. In Section 4.5, we conduct experiments on clean examples to exhibit CMDN's ability to maintain tracking precision while purifying adversarial perturbations. We also deploy CMDN on three additional UAV trackers in Section 4.6 to demonstrate the transferability of defense effectiveness. Section 4.7 compares the processing speed of CMDN with other defense methods to validate its real-time performance. In Section 4.8, we assess the individual contribution of $\mathcal{L}_{rec}$ and the complementary reconstruction process to CMDN's defense effectiveness, and also quantify the impact of the weighting between ξ_1 and ξ_2 on the performance of CMDN. Finally, in Section 4.9, we perform a real-world test to validate the robustness and efficiency of CMDN on a UAV platform. All experimental data are the averages of results obtained from five measurements.

4.1. Implement Details

We implement CMDN with PyTorch and perform our experiments on NVIDIA RTX 3090 GPU with 32 GB RAM. SiamRPN++ [4] is chosen as the victim tracker, and its parameters are frozen during the training phase. The defense network is trained based on the MAE-pre-trained ViT-B/16 (https://dl.fbaipublicfiles.com/mae/visualize/mae_visualize_vit_base.pth (accessed on 21 October 2024)) and on the subset that includes 100000 template-search pairs, which are randomly selected from the ImageNet VID [39] and COCO datasets [40]. We fix the parameter of SiamRPN++ and update the encoder's parameters θ and the decoder's parameters ϕ in the defense network using the AdamW optimizer. Color jitter, horizontal flip, scale jitter, and position jitter are used for augmentation. Before being input into the defense network, the tracking template and search regions are resized to 112×112 and 224×224 pixels, respectively, for both training and inference phase. The setting of training hyperparameters is shown in Table 1.

Since both the search regions and tracking template are susceptible to adversarial attacks, it is necessary to reconstruct both of them during the inference phase. Specifically, while the search region X is reconstructed by inputting the pair (Z, X) into the defense network, the tracking template is reconstructed using (Z, Z) exclusively during the inference phase.

Table 1. A summary of hyperparameter settings for training.

Setting of Hyperparameters	Values
Learning rate γ	1×10^{-4}
AdamW's β_1, β_2	0.9, 0.95
Weight decay λ	0.05
Batch size B	16
Training epochs E	10
Loss weight ξ_1, ξ_2	1, 1

4.2. Testing Datasets and Metrics

The proposed CMDN is evaluated on 3 widely used datasets: UAV123 [22], OTB100 [23], and VOT2018 [24].

Specially, UAV123 is a large-scale UAV benchmark that consists of 123 fully annotated HD video sequences that cover a variety of challenging aerial scenarios, including frequent occlusion, low resolution, out-of-view scenarios, etc. Thus, UAV123 is capable of thoroughly evaluating the performance of UAV tracking. OTB100 contains 100 challenging sequences captured in daily life, each labeled with 9 attributes representing specific difficult scenes, such as illumination variation, scale variation, occlusion, deformation, and motion blur.

These two datasets utilize the one-pass evaluation (OPE) method, which features two main metrics: Success and Precision. Success is determined by the ratio of frames with successful tracking, where the overlap score (the Intersection over Union between the predicted bounding boxes and manually labeled ground truths) exceeds a given threshold.

Precision, on the other hand, is defined as the average Euclidean distance between the center locations of the tracked targets and the corresponding ground truths.

VOT2018 is another challenging tracking benchmark that consists of 60 videos. It assesses tracking performance using a completely different set of metrics from OPE, including accuracy, robustness, and expected average overlap (EAO) [41]. Accuracy is calculated as the average overlap between the predicted and ground truth bounding boxes during successful tracking periods. Robustness measures how many times the tracker loses the target (fails) during tracking. The final evaluation results are ranked by EAO, which is a principled combination of accuracy and robustness.

4.3. Robustness on Non-Adaptive Attacks

Generally, the approaches for evaluating the robustness of defense methods can be categorized into two primary types: non-adaptive attacks and adaptive attacks. In a non-adaptive attack scenario, adversarial attacks are directly applied to the victim model before the defense process. Recently, a number of studies [42–44] have demonstrated that non-adaptive attacks are static and could be circumvented, thereby failing to comprehensively assess the effectiveness of defense methods. To address this, researchers have introduced the concept of adaptive attacks. In this scenario, attackers can dynamically adjust their strategies by retraining the attack model or iteratively optimizing perturbations on defense examples. Adaptive attacks are now recognized as an essential criterion when assessing the robustness of defense methods. Therefore, we conduct experiments against non-adaptive attacks to preliminarily evaluate the defense effectiveness of CMDN in this section, and further assess its performance against adaptive attacks in Section 4.4.

To investigate the generalization of CMDN against various adversarial attacks in non-adaptive attack scenarios, we employ the white-box attack methods CSA and Ad^{2}Attack and the black-box attack method IoU Attack to generate adversarial perturbations. Since CSA is capable of attacking both a template branch and a search branch, we conduct experiments on UAV123, OTB100, and VOT2018 in three defense patterns: template branch only, search branch only, and both branches. The results are demonstrated in the column *Non-Adaptive Defense Result* of Table 2.

In terms of the defense effectiveness against CSA, CMDN exhibits an excellent ability to enhance adversarial robustness across all defense patterns on three benchmarks. As illustrated in Exp. No. 1-2, the Success rate and Precision rate have seen a remarkable surge, with increases of 273.38% and 142.50%, respectively. In addition, in Exp. No. 3-1, a significant growth of 132.52% in the EAO metric further demonstrates the performance of CMDN from an alternative perspective in the template defense pattern. Moreover, compared with AADN [17], CMDN exhibits superior performance against non-adaptive CSA across three benchmarks, as shown in Table 3.

With respect to IoU Attack and Ad^{2}Attack, we only consider the search defense pattern, as they merely generate perturbations in the search branch. CMDN demonstrates excellent robustness against these two attack methods across three datasets. For instance, Exp. No. 2-5 and Exp. No. 3-4 show that our defense network can notably improve the tracking performance against both white-box and black-box attacks in the search defense pattern.

Table 2. The defense effectiveness of CMDN against different adversarial attacks on SiamRPN++. *Defense Pattern* represents the deployment position of the defense network. *Attack Result* indicates the original attack results without defense. *Non-Adaptive Defense Result* denotes the defense performance of CMDN against unadjusted adversarial attacks. *Adaptive Defense Result* represents the defense performance in adaptive attack scenarios. $\Delta_{non-adpt\%}$ and $\Delta_{adpt\%}$ denote the enhancement of tracking performance compared with original attack results. The best results are highlighted in green.

Dataset	Exp. No.	Attack Method	Defense Pattern	Metrics	Attack Result	Non-Adaptive Defense Result	$\Delta_{non-adpt\%}$	Adaptive Defense Result	$\Delta_{adpt\%}$
UAV123	1-1	CSA	Template Only	Success ↑	0.478	0.579	0.101 (21.13%)	0.583	0.105 (21.97%)
				Precision ↑	0.656	0.782	0.126 (19.21%)	0.783	0.127 (19.36%)
	1-2		Search Only	Success ↑	0.154	0.575	0.421 (273.38%)	0.551	0.397 (257.79%)
				Precision ↑	0.320	0.776	0.456 (142.50%)	0.747	0.427 (133.44%)
	1-3		Both	Success ↑	0.156	0.569	0.413 (264.74%)	0.550	0.394 (252.56%)
				Precision ↑	0.327	0.776	0.449 (137.31%)	0.759	0.432 (132.11%)
	1-4	IoU Attack	Search Only	Success ↑	0.459	0.552	0.093 (20.26%)	0.568	0.109 (23.75%)
				Precision ↑	0.595	0.749	0.154 (25.88%)	0.757	0.162 (27.23%)
	1-5	Ad^2Attack	Search Only	Success ↑	0.343	0.564	0.221 (64.43%)	0.541	0.198 (57.73%)
				Precision ↑	0.501	0.777	0.276 (55.09%)	0.758	0.257 (51.30%)
OTB100	2-1	CSA	Template Only	Success ↑	0.527	0.628	0.101 (19.17%)	0.625	0.098 (18.60%)
				Precision ↑	0.713	0.834	0.121 (16.97%)	0.829	0.116 (16.27%)
	2-2		Search Only	Success ↑	0.349	0.627	0.278 (79.66%)	0.614	0.265 (75.93%)
				Precision ↑	0.491	0.836	0.345 (70.26%)	0.824	0.333 (67.82%)
	2-3		Both	Success ↑	0.324	0.624	0.300 (92.59%)	0.616	0.292 (90.12%)
				Precision ↑	0.471	0.835	0.364 (77.28%)	0.825	0.354 (75.16%)
	2-4	IoU Attack	Search Only	Success ↑	0.499	0.603	0.104 (20.84%)	0.613	0.114 (22.85%)
				Precision ↑	0.644	0.800	0.156 (24.22%)	0.817	0.173 (26.86%)
	2-5	Ad^2Attack	Search Only	Success ↑	0.259	0.459	0.200 (77.22%)	0.442	0.183 (70.66%)
				Precision ↑	0.315	0.636	0.321 (101.90%)	0.623	0.308 (97.78%)

Table 2. *Cont.*

Dataset	Exp. No.	Attack Method	Defense Pattern	Metrics	Attack Result	Non-Adaptive Defense Result	$\Delta_{non-adpt\%}$	Adaptive Defense Result	$\Delta_{adpt\%}$
VOT2018	3-1	CSA	Template Only	EAO ↑	0.123	0.286	0.163 (132.52%)	0.266	0.143 (116.26%)
				Accuracy ↑	0.541	0.599	0.058 (10.72%)	0.600	0.059 (10.91%)
				Robustness ↓	1.147	0.421	0.726 (63.30%)	0.529	0.618 (53.88%)
	3-2		Search Only	EAO ↑	0.073	0.261	0.188 (257.53%)	0.250	0.177 (242.47%)
				Accuracy ↑	0.486	0.589	0.103 (21.19%)	0.583	0.097 (19.96%)
				Robustness ↓	2.074	0.529	1.545 (74.49%)	0.567	1.507 (72.66%)
	3-3		Both	EAO ↑	0.073	0.248	0.175 (239.73%)	0.234	0.161 (220.55%)
				Accuracy ↑	0.467	0.583	0.116 (24.84%)	0.591	0.124 (26.55%)
				Robustness ↓	2.013	0.515	1.498 (74.42%)	0.632	1.381 (68.60%)
	3-4	IoU Attack	Search Only	EAO ↑	0.129	0.184	0.055 (42.64%)	0.195	0.066 (51.16%)
				Accuracy ↑	0.568	0.583	0.015 (2.64%)	0.591	0.023 (4.05%)
				Robustness ↓	1.171	0.810	0.361 (30.83%)	0.740	0.431 (36.81%)
	3-5	$Ad^2Attack$	Search Only	EAO ↑	0.103	0.163	0.06 (58.25%)	0.154	0.051 (49.51%)
				Accuracy ↑	0.434	0.532	0.098 (22.58%)	0.530	0.096 (22.12%)
				Robustness ↓	1.428	0.894	0.534 (37.39%)	0.985	0.443 (31.02%)

Table 3. The defense effectiveness of CMDN against CSA on SiamRPN++ vs. AADN. *Attack Result* indicates the original attack results without defense. *Non-Adaptive Defense Result* denotes the defense performance of CMDN against unadjusted adversarial attacks. *Adaptive Defense Result* represents the defense performance in adaptive attack scenarios. $\Delta_{non-adpt\%}$ and $\Delta_{adpt\%}$ denote the enhancement of tracking performance compared with the original attack results. The best and the second-best results are highlighted in green and red, respectively.

Dataset	Defense Method	Metrics	Attack Result	Non-Adaptive Defense Result	$\Delta_{non-adpt\%}$	Adaptive Defense Result	$\Delta_{adpt\%}$
UAV123	AADN	Success ↑	0.156	0.525	0.369 (236.54%)	0.476	0.320 (205.13%)
		Precision ↑	0.327	0.722	0.395 (120.80%)	0.678	0.351 (107.34%)
	CMDN	Success ↑	0.156	0.569	0.413 (264.74%)	0.550	0.394 (252.56%)
		Precision ↑	0.327	0.776	0.449 (137.31%)	0.759	0.432 (132.11%)
OTB100	AADN	Success ↑	0.324	0.559	0.235 (72.53%)	0.403	0.079 (24.38%)
		Precision ↑	0.471	0.777	0.306 (64.97%)	0.573	0.102 (21.66%)
	CMDN	Success ↑	0.324	0.624	0.300 (92.59%)	0.616	0.292 (90.12%)
		Precision ↑	0.471	0.835	0.364 (77.28%)	0.825	0.354 (75.16%)
VOT2018	AADN	EAO ↑	0.073	0.14	0.067 (91.78%)	0.109	0.036 (49.32%)
		Accuracy ↑	0.486	0.546	0.079 (16.91%)	0.488	0.021 (4.50%)
		Robustness ↓	2.074	1.063	0.95 (47.19%)	1.395	0.618 (30.70%)
	CMDN	EAO ↑	0.073	0.248	0.175 (239.73%)	0.234	0.161 (220.55%)
		Accuracy ↑	0.486	0.583	0.116 (24.84%)	0.591	0.124 (26.55%)
		Robustness ↓	2.074	0.515	1.498 (74.42%)	0.632	1.381 (68.60%)

4.4. Robustness on Adaptive Attacks

In this section, we introduce the process and results of experiments in adaptive attack scenarios. For CSA and Ad^{2}Attack, we combine the victim tracker with our defense network to form an integrated network. Subsequently, the two attack models are retrained on this integrated network to obtain the adaptive attack models. Regarding the IoU Attack, we query the integrated network and calculate the gradient information to optimize the adversarial perturbations.

Numerous experiment results in Table 2 illustrate that CMDN is capable of maintaining outstanding defense effectiveness when encountering adaptive attacks. As shown in Exp. No. 1-3, the Success rate and Precision rate still increase by 252.56% and 132.11%, respectively, despite that the CSA perturbation generator is pre-trained with the knowledge of the entire defense network. Moreover, in Exp. Nos. 1-1, 3-1, and 3-3, some metrics from the defense results in adaptive attack scenarios even surpass those in non-adaptive attack scenarios, which further validates the robustness of CMDN.

Specially, in Exp. Nos. 1-4, 2-4, and 3-4, we notice that CMDN illustrates superior performance in adaptive attack scenarios compared with non-adaptive ones. AADN [17] also encountered this situation, which is attributed to the attack mechanism of IoU Attack. Concretely, IoU Attack queries the integrated network to calculate perturbations, and thus the querying results are based on defense examples rather than clean ones. Subsequently, these optimized perturbations are added to clean examples, which means that a portion of the perturbations might lose their effectiveness. Consequently, the tracking robustness, on the contrary, could be enhanced to a certain extent.

In addition, Table 3 illustrates that CMDN outperforms AADN in terms of robustness against adaptive CSA across three benchmarks, which is attributed to its capability to effectively eliminate adversarial perturbations while maintaining tracking precision through high-quality reconstruction guided by pre-trained visual representations.

4.5. Performance on Clean Examples

In real-world scenarios, users are entirely unaware if the real-time video frames have been injected with adversarial perturbations. In order to deal with the potential adversarial attacks that could occur at any moment, the defense mechanism must be activated by default. Therefore, the defense network is tasked not only with purifying potential adversarial perturbations but also with avoiding significant degradation of performance on clean video frames. To evaluate the impact of CMDN on the tracking performance on clean examples, we conduct experiments on three datasets, and the results are presented in Table 4.

Table 4. The tracking performance of SiamRPN++ on clean examples when deployed with CMDN. *Original Result* denotes the tracking performance on clean examples, and *Defense Result* represents the tracking performance on defense examples. Δ_{def} denotes the degradation of tracking performance between *Original Result* and *Defense Result*.

Dataset	Metrics	Original Result	Defense Result	Δ_{def} (%)
UAV123	Success ↑	0.611	0.578	−0.033 (−5.40%)
	Precision ↑	0.804	0.776	−0.028 (−3.48%)
OTB100	Success ↑	0.695	0.644	−0.051 (−7.34%)
	Precision ↑	0.905	0.856	−0.049 (−5.41%)
VOT2018	EAO ↑	0.352	0.283	−0.069 (−19.60%)
	Accuracy ↑	0.601	0.597	−0.004 (−0.67%)
	Robustness ↓	0.29	0.393	−0.103 (−35.52%)

In terms of UAV123 and OTB100 benchmarks, employing OPE, our defense network introduces a maximum decrease of 7.34% in the Success rate and 5.41% in the Precision rate. These reductions are considered to be relatively minor. In terms of VOT2018, CMDN exhibits excellent accuracy, and the 19.6% reduction in EAO is mainly due to an increase in robustness, which reflects the frequency of target loss events.

4.6. Transferability on Different UAV Trackers

In the previous sections, we have assessed the effectiveness of CMDN on SiamRPN++, which is an SNN-based tracker. However, there are ViT-based trackers that extract and encode visual features from video frames in a manner notably distinct from SNN-based approaches, as they have entirely different backbones. Therefore, further experiments should be conducted on heterogeneous trackers to comprehensively prove the transferability of our defense network.

We directly apply CMDN to three additional UAV trackers, SiamAPN [5], HiFT [7], and Aba-ViTrack [34], without retraining. Concretely, SiamAPN is an SNN-based two-stage tracking method with a no-prior adaptive anchor proposal network. HiFT proposes a hierarchical feature transformer to learn relationships among multi-level features, and this module is placed between a CNN feature extraction network and a classification and regression network. Aba-ViTrack integrates the feature learning and template-search coupling into an efficient one-stream ViT for real-time UAV tracking. The detailed experimental results in UAV123 are demonstrated in Table 5.

Concretely, we employ Ad^{2}Attack to attack SiamAPN and HiFT, and utilize IoU Attack to attack Aba-ViTrack, as other attack methods are less effective against ViT-based trackers. The experimental results illustrate that, even without retraining, CMDN is capable of boosting the robustness on three heterogeneous UAV trackers. As demonstrated in the defense result on SiamAPN, the tracking performance gains a significant increase of 251.80% in Success and 102.33% in Precision. The result on HiFT and Aba-ViTrack proves that CMDN is also applicable to a ViT-based tracker, which has a completely different architecture from the training baseline tracker SiamRPN++.

Table 5. The tracking performance of three additional UAV trackers in UAV123, where CMDN is deployed without retraining. Δ_{def} represents the enhancement of tracking performance between *Attack Result* and *Defense Result*.

Victim Tracker	Attack Method	Metrics	Original Result	Attack Result	Defense Result	Δ_{def} (%)
SiamAPN	Ad^{2}Attack	Success ↑	0.575	0.139	0.489	0.350 (251.80%)
		Precision ↑	0.765	0.343	0.694	0.351 (102.33%)
HiFT	Ad^{2}Attack	Success ↑	0.589	0.263	0.473	0.210 (79.85%)
		Precision ↑	0.787	0.399	0.652	0.253 (63.41%)
Aba-ViTrack	IoU Attack	Success ↑	0.664	0.581	0.630	0.049 (8.43%)
		Precision ↑	0.864	0.805	0.832	0.027 (3.35%)

The excellent transferability of CMDN is attributed to its unique defensive mechanism. First, $\mathcal{L}_{pix}$ is tracker-agnostic, enabling CMDN to reconstruct video frames regardless of the victim tracker. Furthermore, the reconstruction process is notably robust. By employing two forward passes that involve random masking, reconstruction, and concatenation, the adversarial perturbations are probably disrupted and will lose their effectiveness. These key features enable CMDN to transfer its defense performance to various heterogeneous trackers without retraining.

4.7. Test of Defense Efficiency

The defense network should not significantly increase computational overhead when integrated with trackers. Thus, we conduct experiments to compare CMDN's defense efficiency with other defense methods on OTB100. As illustrated in Table 6, the original SiamRPN++ tracker processes a frame in 9.35 ms, while LRR introduces an additional overhead of 29.11 ms/frame, which substantially reduces the tracking speed. In contrast, although CMDN is slightly less efficient than AADN, it still exhibits outstanding real-time performance, leading to only an 8.83 ms/frame increase in inference time. Therefore, CMDN is capable of considerably enhancing the robustness of UAV trackers against adversarial attacks, while maintaining excellent tracking speed.

Table 6. The speed performance of SiamRPN++ integrated with different defense methods in OTB100.

Defense Method	Frame per Second	Cost per Frame (ms)	Δ_t
Original SiamRPN++	107	9.35	-
AADN	68	14.71	−5.36
CMDN	55	18.18	−8.83
LRR	26	38.46	−29.11

4.8. Ablation Studies

In this work, our approach introduces two main modifications to the original MAE: (a) reconstruction loss function $\mathcal{L}_{rec}$ and (b) complementary reconstruction process. Therefore, we conduct experiments to investigate the individual contributions of these two mechanisms to the effectiveness of CMDN against CSA.

As demonstrated in Table 7, in Exp. No. 2, the introduction of $\mathcal{L}_{rec}$ enhances the adversarial robustness of the defense network. However, in Exp. No. 3, the integration of our reconstruction process even leads to a reduction in defense effectiveness compared with the original MAE model in Exp. No. 1. We suggest that this reduction is due to the loss of too many key visual features during the complete reconstruction of video frames.

Table 7. The tracking performance of SiamRPN++ in the OTB100 dataset against CSA when deployed with different types of defense networks. $MAE_{50\%}$ refers to the original MAE model with a masking ratio of 50%. $MAE_rec_{50\%}$ represents the MAE model trained by $\mathcal{L}_{rec}$ with a masking ratio of 50%. MAE_comp denotes the original MAE model that employs the complementary reconstruction process. The subscripts in $CMDN_{(0.5,1)}$ and $CMDN_{(1,0.5)}$ represent the settings of ξ_1 and ξ_2. The best results are highlighted in green.

Exp. No.	Network Type	Metircs	Attack Result	Defense Result	Δ (%)
1	$MAE_{50\%}$	Success Precision	0.324 0.471	0.576 0.775	0.252 (77.78%) 0.304 (64.54%)
2	$MAE_rec_{50\%}$	Success Precision	0.324 0.471	0.608 0.805	0.284 (87.65%) 0.334 (70.91%)
3	MAE_comp	Success Precision	0.324 0.471	0.538 0.718	0.214 (66.05%) 0.247 (52.44%)
4	CMDN	Success Precision	0.324 0.471	0.624 0.835	0.300 (92.59%) 0.364 (77.28%)
5	$CMDN_{(0.5,1)}$	Success Precision	0.324 0.471	0.582 0.784	0.258 (79.63%) 0.313 (66.45%)
6	$CMDN_{(1,0.5)}$	Success Precision	0.324 0.471	0.611 0.810	0.287 (88.58%) 0.339 (71.97%)

To address this issue, CMDN combines these two mechanisms to achieve the best defense effectiveness among the four defense networks, as shown in Exp. No. 4. Concretely, $\mathcal{L}_{rec}$ is designed to preserve useful visual features for object tracking, thereby addressing the defect observed in Exp. No. 3. Additionally, the complementary reconstruction process further improves the purification of adversarial perturbations. As a result, CMDN demonstrates outstanding defense effectiveness against adversarial examples and maintains excellent reconstruction quality on clean examples, thus ensuring superior adversarial robustness and tracking precision for UAV tracking.

Additionally, we conduct experiments to measure the impact of the weighting between ξ_1 and ξ_2 on the defense effectiveness of CMDN. As shown in Exp. Nos. 4, 5, and 6, assigning excessive weight to either $\mathcal{L}_{pix}$ or $\mathcal{L}_{feat}$ could cause a degradation in defense performance. Specifically, neglecting the learning of representations in the pixel space (reduce ξ_1) will lead to a significant reduction in the reconstruction quality of video frames, which in turn may result in tracking failure. Further, weakening the learning of crucial tracking features (reduce ξ_2) will make the defense network less effective against adversarial attacks. Consequently, we choose to balance the influence of these two loss functions on CMDN during the training phase.

4.9. Real-World Tests and Visualization

As demonstrate in Figure 7, CMDN is further deployed on a UAV platform to validate its effectiveness and real-time performance in real-world scenarios.

In terms of hardware environment, the platform is deployed on an AmovLab P450 Drone [45], equipped with an Intel RealSense D435i camera capable of capturing 1920 × 1080 RGB images, and an NVIDIA Jetson Orin 16 GB embedded board (25 W mode) to provide computational support.

The software environment consists of Ubuntu 20.04 as the operating system and the Robot Operating System (ROS) as the middleware for handling inter-module interactions, and the functionality invocation of the drone includes flight control, image acquisition, and visualization result publishing.

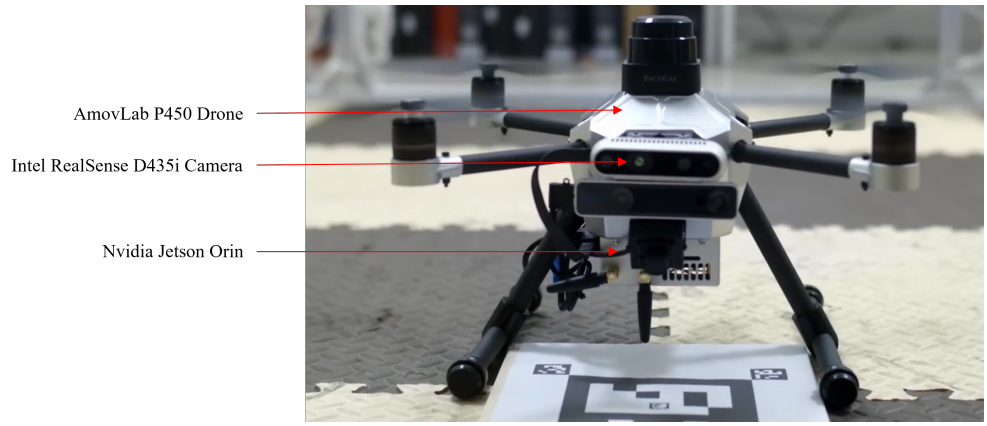


Figure 7. The UAV platform employed in real-world tests.

We conduct the real-world testing of SiamAPN [5] integrated with CMDN against Ad^{2}Attack. As illustrated in Figure 8, our defense network still demonstrates reliable robustness on the UAV platform, including occlusions (#74 to #167 in the second row), deformations (#30 to #108 in the third row), and size changes (#108 to #202 in the third row). In addition, as shown in Table 8, CMDN with SiamAPN achieves 27 frames per second on the NVIDIA Jetson Orin 16 GB (25 W mode), fulfilling the requirements for real-time tracking.

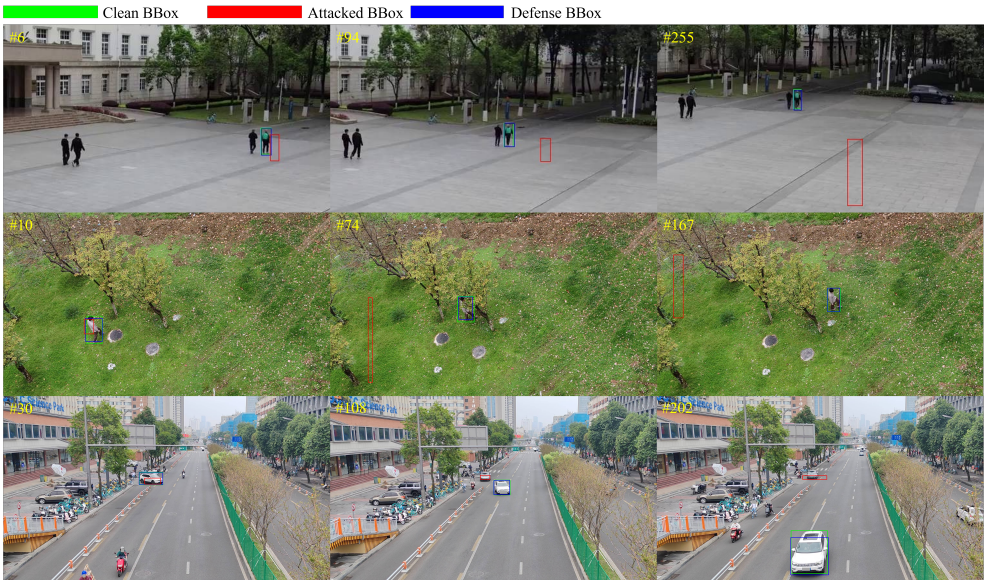


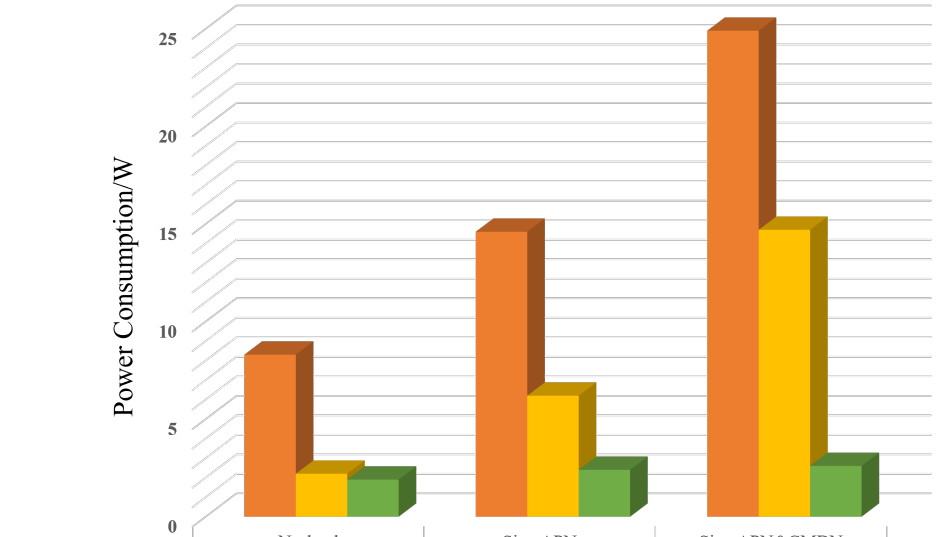
Figure 8. The visualization results of the real-world test on the UAV platform. The yellow numbers represent the indices of video frames.

Table 8. The speed performance of SiamAPN integrated with CMDN on an UAV platform.

Defense Method	Frame per Second	Cost per Frame (ms)	Δ_t
Original SiamAPN	65	15.38	-
CMDN	27	37.04	−21.66

Additionally, we employ the jtop toolkit to monitor the power consumption of the UAV platform during three distinct stages in the real-world test: the no-load stage, the SiamAPN stage, and the SiamAPN & CMDN stage. As depicted in Figure 9, during the no-load stage, the UAV platform operates only essential functional modules and the camera, with a total power consumption of 8.3 W. With the activation of SiamAPN, the power consumption increases to 14.6 W. Furthermore, when both SiamAPN and CMDN are engaged, the power

consumption peaks at 24.9 W, reaching the upper limit of the power threshold we have established. Between the SiamAPN stage and the SiamAPN and CMDN stage, the majority of the additional power consumption is attributed to CPU and GPU, which constitutes 82.5% of the total increase.



	No-load	SiamAPN	SiamAPN&CMDN
VDD_IN	8.3	14.6	24.9
VDD_CPU_GPU_CV	2.2	6.2	14.7
VDD_SOC	1.9	2.4	2.6

Figure 9. The average power consumption of the UAV platform at three stages in the real-world test. *VDD_IN* represents the power consumption of the entire system. *VDD_CPU_GPU_CV* represents the power consumption of CPU and GPU. *VDD_SOC* represents the power consumption of the internal SOC processor.

5. Discussion

So far, we have proved that CMDN is capable of enhancing the robustness of UAV trackers against adversarial attacks in both non-adaptive and adaptive scenarios, and the real-world tests also demonstrate its stable performance and applicable efficiency on a UAV platform, which is significantly beneficial for addressing the challenges in UAV tracking against adversarial attacks.

However, there is still some room for improvement in tracking precision on clean examples and inference speed on UAV platforms, as shown in Tables 4 and 8. To address the issue of precision, CMDN's reconstruction quality must be further upgraded. This could be achieved by revising the loss functions to learn visual representations more effectively or by modifying the reconstruction process, such as refining the reconstruction granularity of those key regions in video frames. Regarding the issue of efficiency, approaches such as employing more lightweight architectures or pruning less critical network layers will reduce the complexity of the defense network. This reduction could enhance real-time performance and decrease power consumption when the network is deployed on a UAV platform. In addition, the balance between processing efficiency and defense effectiveness must be reconsidered carefully. These approaches are worthwhile to deeply investigate in the near future.

6. Conclusions

In this work, we propose a novel adversarial defense network, CMDN, designed to enhance the robustness of UAV tracking and to highlight the vulnerability of UAV trackers to the adversarial examples. Extensive experimental results illustrate that CMDN exhibits remarkable defense effectiveness against various adversarial attacks in both adaptive and non-adaptive attack scenarios, achieving up to 273.38% improvement in tracking

performance. Additionally, CMDN demonstrates excellent transferability across UAV trackers with diverse architectures. Real-world tests validate its reliable robustness and real-time performance on a UAV platform, with CMDN attaining 27 FPS on NVIDIA Jetson Orin 16 GB (25 W mode). Future studies will mainly focus on enhancing the reconstruction quality and compressing the model complexity to improve the network's effectiveness and efficiency on a UAV platform. Overall, our contributions are expected to foster further advancements in UAV tracking applications and bolster ongoing efforts in the field of adversarial robustness.

Author Contributions: Conceptualization, R.Y. and Z.W.; data curation, R.Y.; formal analysis, R.Y.; funding acquisition, S.Z. and Q.L.; investigation, R.Y.; methodology, R.Y. and Q.L.; project administration, Q.L. and S.Z.; supervision, S.Z.; validation, R.Y. and Q.L.; visualization, R.Y., M.G. and B.X.; writing—original draft, R.Y. and Q.L.; writing—review and editing, R.Y., Z.W., M.G. and B.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Open Project of the Intelligent Terminal Key Laboratory of Sichuan Province (SCITLAB-30003) and the National Natural Science Foundation of China (62272089).

Data Availability Statement: Publicly available datasets were analyzed in this study. The UAV123 dataset is available in [22]. The OTB100 dataset is available in [23]. The VOT2018 dataset is available in [24].

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Morando, L.; Recchiuto, C.T.; Calla, J.; Scuteri, P.; Sgorbissa, A. Thermal and visual tracking of photovoltaic plants for autonomous UAV inspection. *Drones* **2022**, *6*, 347. [CrossRef]
2. Xie, X.; Xi, J.; Yang, X.; Lu, R.; Xia, W. Stftrack: Spatio-temporal-focused siamese network for infrared uav tracking. *Drones* **2023**, *7*, 296. [CrossRef]
3. Gao, Z.; Li, D.; Wen, G.; Kuai, Y.; Chen, R. Drone based RGBT tracking with dual-feature aggregation network. *Drones* **2023**, *7*, 585. [CrossRef]
4. Li, B.; Wu, W.; Wang, Q.; Zhang, F.; Xing, J.; Yan, J. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4282–4291.
5. Fu, C.; Cao, Z.; Li, Y.; Ye, J.; Feng, C. Onboard real-time aerial tracking with efficient Siamese anchor proposal network. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *60*, 1–13. [CrossRef]
6. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. SiamAPN++: Siamese attentional aggregation network for real-time UAV tracking. In Proceedings of the 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Prague, Czech Republic, 27 September–1 October 2021; pp. 3086–3092.
7. Cao, Z.; Fu, C.; Ye, J.; Li, B.; Li, Y. Hift: Hierarchical feature transformer for aerial tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 15457–15466.
8. Szegedy, C. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
9. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
10. Guo, Q.; Xie, X.; Juefei-Xu, F.; Ma, L.; Li, Z.; Xue, W.; Feng, W.; Liu, Y. Spark: Spatial-aware online incremental attack against visual tracking. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020. Springer: Berlin/Heidelberg, Germany, 2020; pp. 202–219.
11. Jia, S.; Ma, C.; Song, Y.; Yang, X. Robust tracking against adversarial attacks. In Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; Proceedings, Part XIX 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 69–84.
12. Jia, S.; Song, Y.; Ma, C.; Yang, X. Iou attack: Towards temporally coherent black-box adversarial attack for visual object tracking. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 6709–6718.
13. Jiang, Y.; Yin, G. Attention-Enhanced One-Shot Attack against Single Object Tracking for Unmanned Aerial Vehicle Remote Sensing Images. *Remote. Sens.* **2023**, *15*, 4514. [CrossRef]
14. Yan, B.; Wang, D.; Lu, H.; Yang, X. Cooling-shrinking attack: Blinding the tracker with imperceptible noises. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 990–999.
15. Suttapak, W.; Zhang, J.; Zhang, L. Diminishing-feature attack: The adversarial infiltration on visual tracking. *Neurocomputing* **2022**, *509*, 21–33. [CrossRef]
16. Fu, C.; Li, S.; Yuan, X.; Ye, J.; Cao, Z.; Ding, F. Ad 2 attack: Adaptive adversarial attack on real-time uav tracking. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 5893–5899.

17. Wu, Z.; Yu, R.; Liu, Q.; Cheng, S.; Qiu, S.; Zhou, S. Enhancing Tracking Robustness with Auxiliary Adversarial Defense Networks. *arXiv* **2024**, arXiv:2402.17976.
18. Chen, J.; Ren, X.; Guo, Q.; Juefei-Xu, F.; Lin, D.; Feng, W.; Ma, L.; Zhao, J. LRR: Language-Driven Resamplable Continuous Representation against Adversarial Tracking Attacks. *arXiv* **2024**, arXiv:2404.06247.
19. Peng, Z.; Dong, L.; Bao, H.; Ye, Q.; Wei, F. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv* **2022**, arXiv:2208.06366.
20. He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; Girshick, R. Masked autoencoders are scalable vision learners. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 16000–16009.
21. Xie, Z.; Zhang, Z.; Cao, Y.; Lin, Y.; Bao, J.; Yao, Z.; Dai, Q.; Hu, H. Simmim: A simple framework for masked image modeling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 9653–9663.
22. Mueller, M.; Smith, N.; Ghanem, B. A benchmark and simulator for UAV tracking. In Proceedings of the Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part I 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 445–461.
23. Wu, Y.; Lim, J.; Yang, M.H. Online object tracking: A benchmark. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 23–28 June 2013; pp. 2411–2418.
24. Kristan, M.; Leonardis, A.; Matas, J.; Felsberg, M.; Pflugfelder, R.; ˇCehovin Zajc, L.; Vojir, T.; Bhat, G.; Lukezic, A.; Eldesokey, A.; et al. The sixth visual object tracking vot2018 challenge results. In Proceedings of the European Conference on computer VISION (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
25. Dosovitskiy, A. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
26. Bertinetto, L.; Valmadre, J.; Henriques, J.F.; Vedaldi, A.; Torr, P.H. Fully-convolutional siamese networks for object tracking. In Proceedings of the Computer Vision–ECCV 2016 Workshops, Amsterdam, The Netherlands, 8–10+15–16 October 2016; Proceedings, Part II 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 850–865.
27. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *60*, 84–90. [CrossRef]
28. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
29. Vaswani, A. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**. Available online: https://user.phil.hhu.de/~cwurm/wp-content/uploads/2020/01/7181-attention-is-all-you-need.pdf (accessed on 21 October 2024).
30. Khan, S.; Naseer, M.; Hayat, M.; Zamir, S.W.; Khan, F.S.; Shah, M. Transformers in vision: A survey. *ACM Comput. Surv. (CSUR)* **2022**, *54*, 1–41. [CrossRef]
31. Han, K.; Wang, Y.; Chen, H.; Chen, X.; Guo, J.; Liu, Z.; Tang, Y.; Xiao, A.; Xu, C.; Xu, Y.; et al. A survey on vision transformer. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *45*, 87–110. [CrossRef]
32. Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; Hsieh, C.J. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 13937–13949.
33. Yin, H.; Vahdat, A.; Alvarez, J.M.; Mallya, A.; Kautz, J.; Molchanov, P. A-vit: Adaptive tokens for efficient vision transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 10809–10818.
34. Li, S.; Yang, Y.; Zeng, D.; Wang, X. Adaptive and background-aware vision transformer for real-time uav tracking. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 13989–14000.
35. Deng, A.; Han, G.; Chen, D.; Ma, T.; Liu, Z. Slight aware enhancement transformer and multiple matching network for real-time UAV tracking. *Remote. Sens.* **2023**, *15*, 2857. [CrossRef]
36. Devlin, J. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
37. Brown, T.B. Language models are few-shot learners. *arXiv* **2020**, arXiv:2005.14165.
38. Zhao, H.; Wang, D.; Lu, H. Representation learning for visual object tracking by masked appearance transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 18696–18705.
39. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]
40. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. In Proceedings of the Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, 6–12 September 2014; Proceedings, Part V 13; Springer: Berlin/Heidelberg, Germany, 2014; pp. 740–755.
41. Čehovin, L.; Leonardis, A.; Kristan, M. Visual object tracking performance measures revisited. *IEEE Trans. Image Process.* **2016**, *25*, 1261–1274. [CrossRef] [PubMed]
42. Carlini, N.; Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017; pp. 3–14.
43. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In Proceedings of the International Conference on Machine Learning, PMLR, Stockholm, Sweden, 10–15 July 2018; pp. 274–283.

44. Tramer, F.; Carlini, N.; Brendel, W.; Madry, A. On adaptive attacks to adversarial example defenses. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1633–1645.
45. Amovlab. Available online: https://amovlab.com/product/detail?pid=43 (accessed on 21 October 2024).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.