


Article

A3GT: An Adaptive Asynchronous Generalized Adversarial Training Method

Zeyi He ¹, Wanyi Liu ², Zheng Huang ¹, Yitian Chen ² and Shigeng Zhang ^{2,*} 

¹ Changsha Urban Development Group Co., Ltd., Changsha 410208, China; hezeyi@csudgroup.com (Z.H.); huangzheng@csudgroup.com (Z.H.)

² School of Computer Science and Engineering, Central South University, Changsha 410083, China; liuwanyi2@huawei.com (W.L.); yt_chen@csu.edu.cn (Y.C.)

* Correspondence: sgzhang@csu.edu.cn

Abstract: Adversarial attack methods can significantly improve the classification accuracy of deep learning models, but research has found that although most deep learning models with defense methods still show good classification accuracy in the face of various adversarial attack attacks, the improved robust models have a significantly lower classification accuracy when facing clean samples compared to themselves without using defense methods. This means that while improving the model's adversarial robustness, it is necessary to find a defense method to balance the accuracy of clean samples (clean accuracy) and the accuracy of adversarial samples (robust accuracy). Therefore, in this work, we propose an Adaptive Asynchronous Generalized Adversarial Training (A3GT) method, which is an improvement over the existing Generalist method. It employs an adaptive update strategy without the need for extensive experiments to determine the optimal starting iteration for global updates. The experimental results show that compared with other advanced methods, A3GT can achieve a balance between clean sample classification accuracy and robust classification accuracy while improving the model's adversarial robustness.

Keywords: adversarial attacks; adversarial robustness; adversarial training; deep learning models; multi-task learning



Citation: He, Z.; Liu, W.; Huang, Z.; Chen, Y.; Zhang, S. A3GT: An Adaptive Asynchronous Generalized Adversarial Training Method.

Electronics **2024**, *13*, 4052.
<https://doi.org/10.3390/electronics13204052>

Academic Editor: Fernando De la Prieta Pintado

Received: 31 August 2024
Revised: 6 October 2024
Accepted: 8 October 2024
Published: 15 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

Deep learning technology has achieved remarkable success in numerous fields, changing the traditional production and lifestyle of humans in various fields. These applications span computer vision [1,2], speech recognition [3,4], natural language processing [5,6], and autonomous driving [7–9]. However, with the discovery of adversarial examples [10], it has become evident that these imperceptible perturbations can cause catastrophic effects on the performance of deep learning models. In recent years, researchers have developed numerous methods for generating adversarial examples [11–14]. In practical applications of deep learning models, adversarial attacks can cause models to make high-confidence erroneous predictions, raising significant concerns about the security and reliability of deep learning systems.

Adversarial robustness refers to the ability of machine learning and deep learning models when subjected to adversarial attacks. These attacks involve applying small but meaningful perturbations to input data to mislead the model into making incorrect predictions. The goal of adversarial robustness is to ensure that models can maintain accurate outputs, even when the inputs are perturbed. Current methods to enhance adversarial robustness can be broadly categorized into model-based and data-based approaches. One of the most effective and straightforward defense methods during model training is adversarial training. This technique incorporates adversarial examples into the training process, making the model more robust against perturbations by learning from these perturbed

samples. Adversarial training is widely regarded as one of the most effective and simple methods for improving model robustness [15].

Many works have already confirmed the effectiveness of adversarial training, and many deep learning models with defense mechanisms exhibit good classification accuracy when faced with various adversarial attacks. However, these models show a significant drop in accuracy when handling clean samples. This highlights the need for a method that not only enhances the overall robustness of the model, but also balances clean sample accuracy (clean accuracy) and adversarial sample accuracy (robust accuracy). To address this issue, Wang et al. proposed Generalist [16], which divides model training into two independent parts, training on natural/adversarial datasets separately, and periodically collecting parameters from both parts to form a global learner. However, this method requires extensive experimentation to determine the optimal starting iteration for global updates. To balance clean accuracy and robust accuracy while reducing the model's reliance on fixed hyperparameters, we propose an Adaptive Asynchronous Generalized Adversarial Training (A3GT) method in this study. Experimental results show that, compared to several advanced methods, A3GT achieves a better balance between clean sample classification accuracy and robust classification accuracy, while eliminating the need for preset hyperparameters, significantly improving the model's adversarial robustness.

2. Background

2.1. Adversarial Training

Adversarial training was first proposed in the literature by Madry et al. [17], who formulated the problem as a minimax optimization. The goal is to maximize the adversarial perturbation error while minimizing the distributional error with the original data distribution. The problem can be described by the following equation:

$$\min_{\theta} \mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\max_{\Delta x \in \Omega} L(x + \Delta x, y; \theta) \right], \quad (1)$$

Here, the outer minimization $\min_{\theta} \mathbb{E}$ ensures that the model's predictions with adversarial samples follow the original data distribution, while the inner maximization $\max_{\Delta x \in \Omega} L(x + \Delta x, y; \theta)$ maximizes the adversarial perturbation error within the perturbation set Ω . Adversarial training updates model parameters using adversarial examples during the iterative training process. The process can be summarized as follows:

$$\begin{cases} x^{(t+1)} = \Pi_{\mathbb{B}(x,\epsilon)} \left(x^{(t)} + \alpha \operatorname{sign} \left(\nabla_{x'} \ell_2 \left(x^{(t)}, y; \theta^t \right) \right) \right), \\ \theta^{(t+1)} = \theta^{(t)} - \tau \nabla_{\theta} \mathbb{E} [\ell_1(x, y; \theta^t) + \beta \mathcal{R}(x', x, y; \theta^t)], \end{cases} \quad (2)$$

where α represents the step size, τ represents the learning rate, and \mathcal{R} represents the loss function. ℓ_1 and ℓ_2 denote the objective functions for clean and adversarial samples, respectively. The hyperparameter β balances clean accuracy and robust accuracy. Many defense methods are different in the value of β . For instance, when $\beta = 1$, it represents Projection Gradient Descent (PGD) adversarial training [17]. When $\beta = 0$, it represents normal training. When \mathcal{R} is the KL divergence, it corresponds to the TRADES defense method [18].

Adversarial training aims to improve model robustness against adversarial attacks, but conventional adversarial training often comes at the cost of performance on clean, unperturbed samples. In practical applications, it is crucial to ensure that models not only maintain high classification accuracy under adversarial attacks, but also perform well under normal conditions. Achieving a balance between clean accuracy and robust accuracy involves careful tuning of the model's structure, training strategies, and adversarial training parameters. Designing robust deep learning models requires considering performance across multiple tasks and environments, adapting to potential threats while maintaining effectiveness on clean data. This is a complex and significant research direction that involves a comprehensive understanding and optimization of model adversarial robustness.

Currently, mainstream approaches to this problem fall into two categories:

- Data-centric approaches: Methods like those used by Carmon et al. [19] and Najafi et al. [20] involve adding extra labeled and unlabeled data. Lee et al. [21] and Zhang et al. [22] focus on adjusting perturbation sizes to generate suitable adversarial samples for better model optimization.
- Model-centric approaches: Wang et al. [16] propose simultaneously updating model parameters through normal adversarial training and clean sample training to achieve a balance between robust accuracy and clean accuracy. However, experiments have shown that different data distributions require different synchronized parameters for model training, making fixed parameter methods less flexible.

Ensuring secure and reliable AI applications demands robust deep learning models capable of withstanding adversarial attacks. However, existing methods face limitations in adapting to different data distributions and model architectures with multi-task parameters. Multi-task learning in deep learning is crucial for handling various tasks and data distributions, allowing models to flexibly adapt to different environments and application scenarios. Yet, current robustness defense methods that balance natural and robust accuracy are constrained by static task weights, limiting their performance across different data distributions or model architectures.

In practical applications, the relationship between data distributions and tasks may evolve over time and environment. Therefore, enhancing a model's adaptive capabilities to flexibly adjust its parameters according to new contexts and requirements is a pressing challenge that needs to be addressed.

2.2. Multi-Task Learning

Multi-task learning refers to training a model to learn multiple tasks simultaneously to improve its overall performance. In multi-task learning, the model is designed to handle and learn from multiple tasks at the same time, rather than training separate models for each task independently. The advantage of multi-task learning lies in its ability to enable the model to share knowledge across related tasks, thereby enhancing overall generalization performance. By training on multiple tasks, the model can learn more general and abstract feature representations, which helps improve its adaptability to new tasks.

Assume we have a set of data distributions \mathcal{D} and loss functions ℓ , defined as $\mathcal{A} = \{\mathcal{D}, \ell\}$. The trained models are denoted as $\{\mathcal{M}_a\}_{a=1}^{|\mathcal{A}|}$, with parameters $\theta_{\mathcal{M}_a}$ learned through training. In multi-task learning, the optimal parameters for each task are obtained through the following formula:

$$\bigcup_{a=1}^{|\mathcal{A}|} \theta_{\mathcal{M}_a}^* = \underset{\bigcup_{a=1}^{|\mathcal{A}|} \theta_{\mathcal{M}_a}}{\operatorname{argmin}} \mathbb{E}_{\mathcal{A}} \mathbb{E}_{\mathcal{D}} \ell_a(\mathcal{D}_a; \theta_{\mathcal{M}_a}), \quad (3)$$

where $\ell_a(\mathcal{D}_a; \theta_{\mathcal{M}_a})$ represents the loss function of the model on the dataset \mathcal{D}_a . To enhance model robustness while maintaining classification accuracy on clean original samples, this study adopts a training approach similar to multi-task learning. Although A3GT may intuitively appear similar to multi-task learning, each task in multi-task learning is interconnected and influences one another, whereas in our approach, the two tasks are independent of each other.

3. Methods

This section provides a detailed introduction to the Adaptive Asynchronous Generalized Adversarial Training (A3GT), including its specific details and implementation process. The implementation process involves two basic learners and a global learner. Unlike Generalist [16], which determines the optimal starting iteration for global updates through experimentation, A3GT aims to find an adaptive update strategy. This adaptive strategy ensures that different model architectures and sample distributions can have their

own tailored training approach. Moreover, the A3GT method maintains or even slightly improves the balance between adversarial robustness and classification accuracy on clean samples compared to existing methods.

3.1. A3GT Overview

Most deep learning models improve adversarial robustness through adversarial training, but their accuracy on clean samples significantly declines compared to before. To address this issue, existing methods include adding known labeled and unknown data from a data perspective. The A3GT method follows the overall training framework of Generalist, preserving the model's ability to learn from clean samples. At the same time, we introduce an adaptive update strategy for the global learner to eliminate the method's dependency on preset hyperparameters. The overall framework of the adaptive asynchronous generalized adversarial training designed in this paper is shown in Figure 1. The conventional adversarial training is modified into two parts. One part learns from the original clean sample images to update model parameters as the clean learner, while the other part learns from adversarial sample images to update parameters of the robust learner. Meanwhile, the second part of the learner also generates adversarial samples to be used in subsequent model training. These two learners will merge under certain conditions to form the initial global learner, which is the final deep learning model.

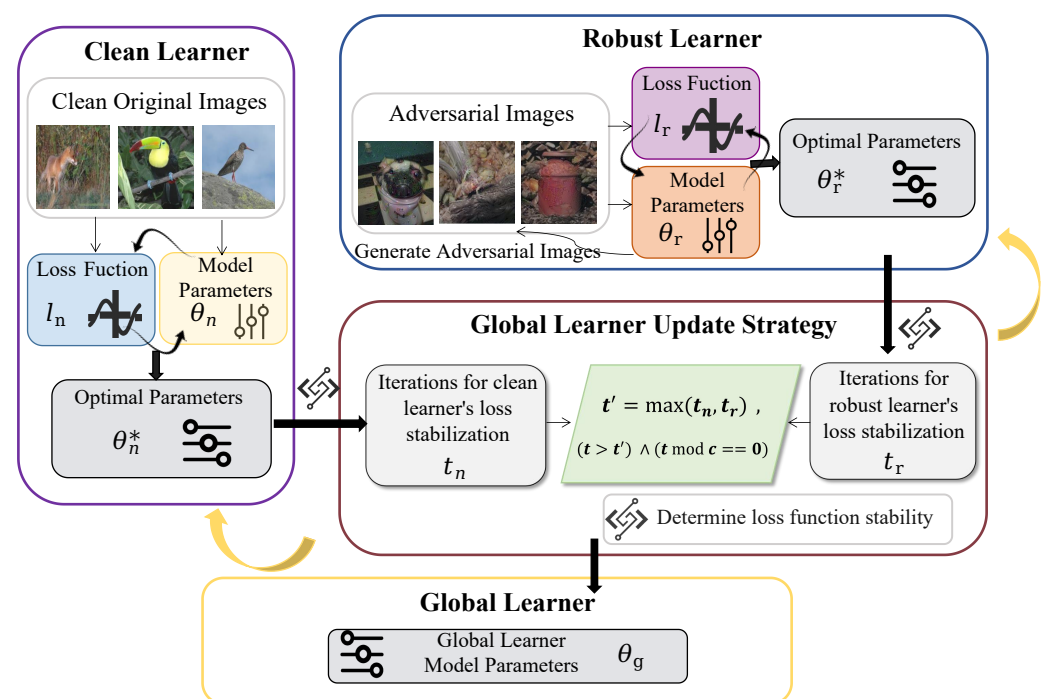


Figure 1. The overall framework of the Adaptive Asynchronous Generalized Adversarial Training.

In this way, the model can maintain adversarial robustness while preserving its ability to learn from clean samples. However, a remaining challenge is determining the optimal timing for merging the two basic learners. Existing methods use fixed interaction iterations determined through experiments, which lack flexibility. The moment when the loss function stabilizes varies depending on different data distributions and model architectures. By assessing whether the two loss functions have stabilized, interactions and parameter transfers occur after both loss functions reach stability to obtain the global learner. It is important to note that after beginning to update the global learner, it is not updated through interactive learning in every iteration. Updating the global learner in every iteration would result in the basic learners not having enough experience, consuming a lot of time. On

the other hand, if the update frequency is too slow, critical learning experiences might be missed. Therefore, determining a reasonable update frequency is crucial.

The formulas for generating the global learner from the clean learner, trained on original clean sample images, and the robust learner, trained on adversarial sample images, are as follows:

$$\theta_g \leftarrow \alpha' \theta_g + (1 - \alpha')(\gamma \theta_r + (1 - \gamma) \theta_n), \quad (4)$$

where θ_g represents the model parameters of the global learner. The parameter update mechanism in the global learner considers two main factors: the global model parameters from the previous stage and the parameters of the two independent basic learners. These two sets of parameters are combined through weighted synthesis for the update, with the constraint that their weights sum to 1. The coefficients α' and γ are the weighting factors.

3.2. Basic Learners

The A3GT method involves collaborative learning and updating by two basic task learners, referred to in this study as the clean learner and the robust learner. We will now formalize these two basic tasks.

Given a global dataset \mathcal{D} , the original clean sample dataset is \mathcal{D}_1 , and the dataset after adversarial attack is \mathcal{D}_2 , where $\mathcal{D}_1 \cup \mathcal{D}_2 \subseteq \mathcal{D}$. A clean sample image with label y is $x \in \mathbb{R}^m$, $(x, y) \sim \mathcal{D}_1$. The DNNs classification model trained on clean samples is denoted as $f_n(x) : \mathbb{R}^m \rightarrow \{1, \dots, k\}$. The objective is to achieve correct classification for clean samples using the loss function ℓ_n , with the target function $f_n(x) = y$, indicating that the sample x is correctly classified as label y by the classification model $f_n(x)$. Therefore, the objective of sub-task one, the normal training of the clean learner, to find the optimal parameters of the classification model $f_n(x)$ is expressed as:

$$\theta_n^* = \operatorname{argmin}_{\mathcal{D}_1} \mathbb{E} \ell_n(\mathcal{D}_1; \theta_n), \quad (5)$$

where ℓ_n is the loss function used to measure the difference between predictions and true labels, and \mathbb{E} represents the expectation operator.

Similarly, sub-task two involves training the model using the adversarial sample dataset \mathcal{D}_2 . When clean samples are subjected to adversarial attacks, adversarial sample images x' are generated, represented as $(x', y) \sim \mathcal{D}_2$. The goal of adversarial attacks is to find a suitable x' through the loss function ℓ_r , achieving $f_r(x') \neq y$ (non-targeted attack) or $f_r(x') = y_{adv}$ (targeted attack). Additionally, the perturbation size of x' is usually constrained by the l_p -norm ball, i.e., $\|x' - x\|_p \leq \epsilon$. The optimal parameters of the robust learner's classification model f_r , trained using adversarial samples, are expressed as:

$$\theta_r^* = \operatorname{argmin}_{\mathcal{D}_2} \mathbb{E} \ell_r(\mathcal{D}_2; \theta_r). \quad (6)$$

As previously mentioned, the two sub-tasks are independent of each other. They operate on their respective datasets, update their own basic learner parameters without interference. The global learner, by learning from the two basic learners, achieves the goal of adversarial training while maintaining the classification accuracy of the model when faced with clean original samples, balancing robust accuracy with clean accuracy.

3.3. Adaptive Global Learner

The two basic learners operate independently during their respective training phases. To combine their advantages, they must be integrated. Therefore, a global learner exists to synthesize the two basic learners under specific conditions. The following describes when the global learner integrates the two basic learners and how this integration is achieved.

In the initial stages of training a deep learning model, the two basic learners have limited learning experience. Since model parameters are randomly initialized, parameter updates can be quite volatile. Sharing parameters at this stage could lead to biased training parameters, because the basic learners do not yet have sufficient learning experience to

support reasonable parameter initialization. To address this, we introduce a round size t' , which indicates the synchronization of the global learner's parameters starting from the t' -th round after training begins. Initially, the two basic learners train independently until their respective loss functions, ℓ_n and ℓ_r , stabilize at round t' , determined as follows:

$$\begin{cases} t_n = \text{argming}(\ell_n(\mathcal{D}_1), t_{num}, l_h; \theta_n), \\ t_r = \text{argming}(\ell_r(\mathcal{D}_2), t_{num}, l_h; \theta_r), \\ t' = \max(t_n, t_r), \end{cases} \quad (7)$$

where $\text{g}\{\cdot\}$ is a function that determines whether the current loss function remains below the threshold l_h for t_{num} consecutive rounds, obtaining the round value when the loss function stabilizes. Once the loss functions of the two basic learners stabilize and meet the above conditions, the global learner's parameters start to synchronize. This ensures that both basic learners have enough learning experience before parameter sharing, avoiding bias during parameter initialization. The largest round among the two sub-tasks is chosen to determine the starting round for updating the global learner, ensuring both basic learners have stabilized.

The specific update strategy for the global learner is shown in Equation (8). When the learning rounds of the basic learners $t < t'$, the clean learner and the robust learner learn independently without interference. When the learning round $t = t'$, the parameters of the two basic learners are mixed in proportion to γ to update the learning parameters for the current round. The previous round's learning experience is iteratively updated with a learning rate α' , as follows: $\theta_g \leftarrow \alpha' \theta_g + (1 - \alpha')(\gamma \theta_r + (1 - \gamma) \theta_n)$. The value of α' determines the importance of the basic learners to the global learner. Adjusting γ can control the model's emphasis on handling clean samples versus adversarial samples. Throughout the process of updating the global learner, the basic learners pass parameters every c rounds. The function $\mathcal{B}(t, t', c)$ is a boolean function that returns 1 if and only if $t > t'$ and $t \bmod c == 0$, otherwise it returns 0. In simple terms, after round t' , the global learner's parameters θ_g are updated every c rounds. When $t \bmod c \neq 0$, the clean learner and the robust learner continue their independent training.

$$\begin{cases} \theta_n^t = \text{argmin}_{\mathbb{E}_{\mathcal{D}_1}} \ell_n(\mathcal{D}_1); \theta_n^{t-1}, \\ \theta_r^t = \text{argmin}_{\mathbb{E}_{\mathcal{D}_2}} \ell_r(\mathcal{D}_2); \theta_r^{t-1}, \\ \theta_g^{t+1} = \alpha' \theta_g^{t-1} + (1 - \alpha') [\gamma \theta_r^t + (1 - \gamma) \theta_n^t], \\ \theta_n^t = \mathcal{B}(t, t', c) \theta_g^{t+1} + (1 - \mathcal{B}(t, t', c)) \theta_n^t, \\ \theta_r^t = \mathcal{B}(t, t', c) \theta_g^{t+1} + (1 - \mathcal{B}(t, t', c)) \theta_r^t. \end{cases} \quad (8)$$

4. Experimental Evaluation

4.1. Experimental Setup

4.1.1. Experimental Environment Configuration

Due to the intensive computational power required for training and testing the models, we conducted the experiments using a GPU server. The basic information of the server is shown in Table 1.

Table 1. Server environment and configuration.

Environment	Configuration
Operating system	Ubuntu Server 22.04 (Canonical Ltd., London, UK)
CPU	Intel(R) Xeon(R) Gold 5118 CPU @ 2.30 GHz (Intel Corporation, Santa Clara, CA, USA)
GPU	NVIDIA GeForce RTX 3090 24 GB (NVIDIA Corporation, Santa Clara, CA, USA)
Memory	314 GB
Development language	Python 3.9 (Python Software Foundation, Wilmington, DE, USA)
Deep learning framework	Pytorch 2.0.1 (Meta Platforms, Inc., Menlo Park, CA, USA)

4.1.2. Dataset Selection

The datasets selected for this study include CIFAR-10 [23], MNIST, and SVHN. CIFAR-10 is widely used in computer vision tasks, consisting of 10 different classes with approximately 6000 images per class at a resolution of 32×32 pixels. MNIST, used for handwritten digit recognition tasks, comprises 60,000 training images and 10,000 test images, each grayscale and sized at 28×28 pixels across 10 classes representing digits from 0 to 9. SVHN dataset, collected from real-world street scenes, consists of larger, typically color images sized at 32×32 pixels.

4.1.3. Baseline Adversarial Defense Methods

This study compares the A3GT method with several baseline methods, including standard adversarial training using PGD [17], Generalist [16], FAT [22], IAT [24], TRADES [18], and YOPO [24]. For clean models, ResNet-18 serves as the base model. For standard adversarial training models, ResNet-18 is also used with the AT [25] attack method, where parameter $\beta = 1$. Generalist method uses PGD as the attack method with parameters $\gamma = [1, 1, 1, 0.4]$. FAT and IAT methods follow the parameter settings as specified in their respective papers.

4.1.4. Adversarial Attack Methods

This study employs Projection Gradient Descent (PGD) [17], Momentum Iterative Attack (MIA) [26], and AutoAttack [25], which includes variations such as $APGD_{ce}$, $APGD_{dlr}$, $APGD_t$, FAB_t , and Square for attacking models.

4.1.5. Evaluation Metrics

- Clean accuracy: classification accuracy of the model on clean samples.
- Robust accuracy: classification accuracy of the model on adversarial samples generated by corresponding attack methods.
- CMMR score [27]: Comprehensive Multi-dimensional Model Robustness (CMMR) score derived from metrics including Acc, ASS, MSE, ALD_2 , and PSNR.

4.2. Performance Comparison under Different Adversarial Attack Methods

To comprehensively demonstrate the performance of A3GT, the experimental results of A3GT based on the ResNet18 model on the CIFAR-10 dataset are shown in Table 2.

Table 2. Comparison of clean accuracy (%) and robust accuracy (%) between A3GT method and other defense methods.

Model	Clean Accuracy	PGD20	PGD100	MIM	$APGD_{ce}$	$APGD_{dlr}$	$APGD_t$	FAB_t	Square
Clean	91.52	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Generalist	89.09	50.01	50.00	52.19	46.53	48.70	46.11	47.32	56.68
FAT	87.72	46.69	46.81	47.03	46.20	47.51	44.88	45.76	52.98
IAT	84.60	40.83	40.87	43.07	37.56	37.95	35.13	36.06	49.30
A3GT	89.11	51.22	51.23	53.23	47.87	50.13	46.69	47.20	57.10

The clean model achieves the highest classification accuracy on clean sample images, reaching 91.52%. However, it lacks the ability to defend against adversarial sample images, resulting in complete failure when facing various adversarial attacks. Its robust accuracy is 0% against any type of adversarial attack. This underscores the need for enhancing the adversarial robustness of models, which is why numerous defense methods have emerged to significantly improve robustness under adversarial attacks. Nevertheless, it is evident that conventional methods for enhancing robustness do not consider the impact on the clean accuracy (classification accuracy on clean samples). For example, the FAT and IAT methods show a substantial improvement in adversarial robustness compared to the clean model. In the case of PGD20, the clean model has a robust accuracy of 0%, whereas the FAT and IAT methods increase the robust accuracy to 46.69% and 40.83%, respectively,

demonstrating significant effectiveness in improving adversarial robustness. However, these models perform poorly on original clean sample images. For instance, the IAT method reduces clean accuracy by 6.92% (from 91.52% to 84.6%) compared to the clean model. Thus, there is a need for a defense method that can enhance robust accuracy without a significant drop in clean accuracy when dealing with original clean sample images. Generalist is one such method that achieves a balance between clean accuracy and robust accuracy, improving robust accuracy while minimizing the reduction in clean accuracy. However, it has a drawback: it uses a synchronous approach for normal and adversarial training. This can lead to suboptimal results as the optimal training times for normal and adversarial training can differ depending on data distribution and model architecture. The A3GT method addresses this issue by allowing interaction between the two basic learners when both reach their optimal states, as analyzed in Section 3.3. Experimental results show that the A3GT method achieves the highest robust accuracy under seven types of adversarial attacks compared to the other three defense methods. However, its performance under FAB_t adversarial attack is not as good as Generalist. The A3GT method strikes a balance between clean accuracy and robust accuracy, performing second only to the clean model on original clean sample images while significantly improving robust accuracy.

4.3. Performance Comparison on Different Datasets

In addition to the CIFAR-10 dataset, we tested the performance of the ResNet-18 model architecture on the MNIST and SVHN datasets. The maximum perturbation strength for adversarial attack methods was set to $\epsilon = 8/255$. The experimental results comparing the A3GT method with other baseline methods are shown in Table 3.

Table 3. Comparison of clean accuracy (%) and robust accuracy (%) between A3GT method and other defense methods.

	Clean Samples	MNIST PGD	MIA	Clean Samples	SVHN PGD	MIA
FAT	98.97	92.26	93.54	93.41	53.26	54.54
TRADES	99.13	94.61	95.13	93.13	54.61	55.13
YOPO	99.19	93.13	93.54	92.19	52.13	55.54
A3GT	99.2	96.13	96.3	94.31	54.13	56.3

The results demonstrate that the A3GT method achieves the best performance on clean samples across both datasets, with accuracies of 99.2% on MNIST and 94.31% on SVHN. Additionally, the A3GT method significantly outperforms other baseline methods under the MIA attack, achieving 96.3% on MNIST and 56.3% on SVHN. The A3GT method also shows the best defense against the PGD attack on the MNIST dataset (96.13%) and ranks second only to the TRADES method in classification accuracy on the SVHN dataset.

To comprehensively explore the trade-off between clean accuracy and robust accuracy of different defense methods, this study further analyzes the performance of clean accuracy under various levels of robust accuracy during training. As shown in Figure 2, we conducted a detailed comparison of the A3GT method with existing methods such as Generalist, FAT, and Madry. Overall, the experimental results indicate that most methods exhibit a decline in clean accuracy as robust accuracy increases. Specifically, the FAT and Madry methods show a significant drop in clean accuracy when the robust accuracy approaches 40%. In contrast, the Generalist method achieves a more balanced result between robust accuracy and clean accuracy. Additionally, the A3GT method achieves a better balance between robust accuracy and clean accuracy compared to the Generalist method. The experiments reveal that the A3GT method can maintain robustness while minimizing the decline in clean accuracy, resulting in a more resilient model against adversarial attacks.

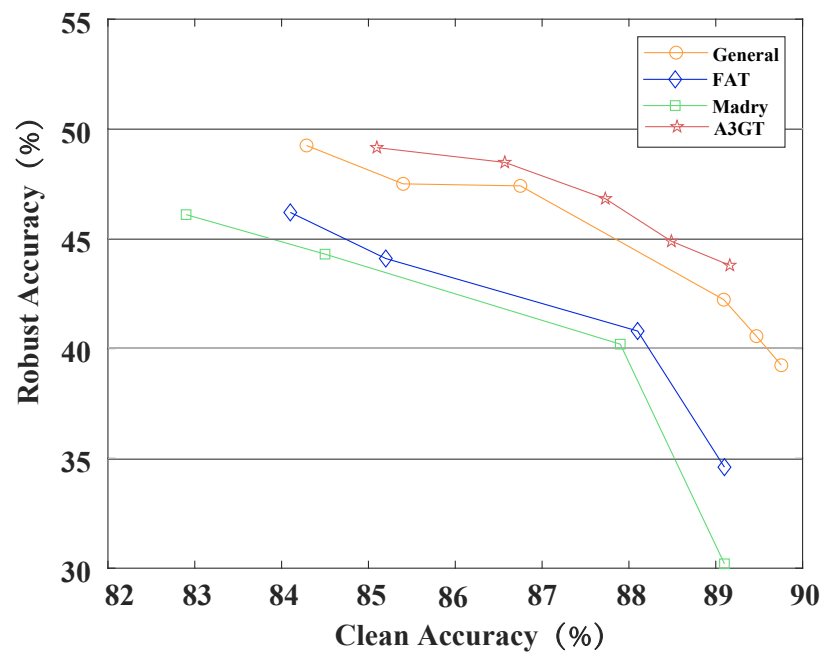


Figure 2. Comparison of clean accuracy and robust accuracy (using AutoAttack adversarial attacks [25]) between the A3GT method and other adversarial training methods.

4.4. Robustness Score of A3GT under the CMMR Framework

In real-world scenarios, the conditions faced by models can vary significantly. To comprehensively and accurately assess the adversarial robustness of models, this paper proposes a Comprehensive Multi-dimensional Model Robustness (CMMR) evaluation framework. In this section, the proposed A3GT method is comprehensively evaluated using the CMMR framework against five other adversarial defense methods: TRADES, YOPO, Self Adaptive, FAT_TRADES_032, and FAT_MART_032. These defense techniques primarily aim to enhance model robustness by altering model parameters during training.

As shown in Figure 3, the CMMR score of the A3GT method generally surpasses that of the other five defense methods. Specifically, when $\epsilon < 0.03$, the A3GT method is slightly inferior to TRADES and Self Adaptive, but the difference is minimal. As the adversarial perturbation strength increases, when $\epsilon < 0.08$, A3GT ranks second only to the FAT_TRADES_032 method, and thereafter, its CMMR score exceeds that of any other defense method. When $\epsilon > 0.13$, the A3GT method is second only to YOPO and Self Adaptive. From these results, it is evident that while the A3GT method does not consistently outperform all other defense methods across the entire range of perturbation strengths, it demonstrates stable and superior performance overall. Additionally, radar charts depicting the CMMR scores of A3GT and the other five defense methods at perturbation strengths of $\epsilon = 0.04$, $\epsilon = 0.08$, $\epsilon = 0.12$, and $\epsilon = 0.16$ are presented in Figure 4. As shown, the A3GT consistently leads in CMMR scores across all four conditions compared to the other five defense methods.

4.5. Ablation Study

In this subsection, a series of ablation studies are conducted on the proposed A3GT method, focusing on the mixing ratio γ of the two basic learners and the learning frequency c of the global learner. These studies aim to explore and analyze how these two parameters affect the performance of the A3GT method.

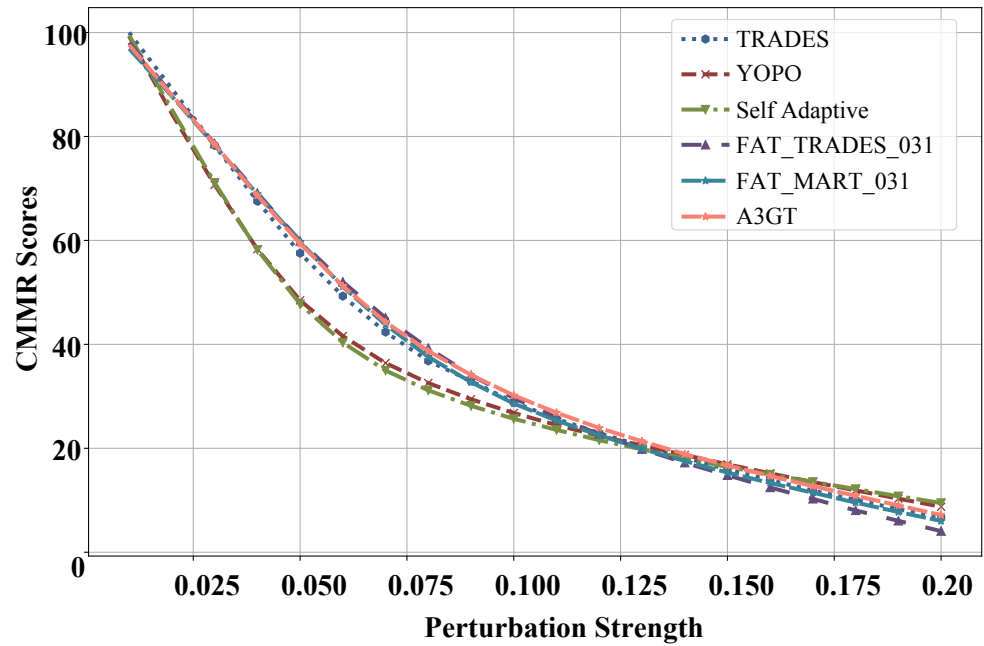


Figure 3. CMMR scores and perturbation strength curve of A3GT method compared with other defense methods.



Figure 4. Comparison of CMMR scores between A3GT method and other defense methods at perturbation strengths of $\epsilon = 0.04, \epsilon = 0.08, \epsilon = 0.12,$ and $\epsilon = 0.16$.

4.5.1. Mixing Ratio γ of the Two Basic Learners

It is important to note that the mixing ratio γ of the two basic learners is not a vector, but a scalar. The model training is divided into different stages, with each stage training the model according to different mixing ratios. The mixing ratio decreases according to a dynamic function as the training iterations increase. As shown in Equation (4), when $\gamma = 1$, the global learner is fully updated by the robust learner. As the mixing ratio gradually decreases, the global learner’s learning shifts from the robust basic learner to the normal basic learner, ensuring that the model retains learning experience from the original clean images.

As illustrated in Figure 5, the effect of different mixing ratios γ on the results is significantly different, and the performance is also influenced by the learning frequency c of

the global learner. In the absence of attacks, the best setting is a mixing ratio $\gamma = (1, 1, 1, 0.4)$ and a learning frequency $c = 5$. Under PGD attacks, the best performance is achieved with a mixing ratio $\gamma = (1, 1, 1, 0.4)$ and a learning frequency $c = 1$. For AutoAttack, the method performs optimally with a mixing ratio $\gamma = (1, 1, 1, 0.4)$, similar to the performance under C&W attacks. Additionally, the experimental results indicate that when the mixing ratio is fixed, the method performs well only with a learning frequency $c = 5$, while the performance significantly degrades under other conditions. Therefore, to enhance the clean accuracy on original clean samples, the A3GT method sets the mixing ratio γ of the two basic learners to $(1, 1, 1, 0.4)$.

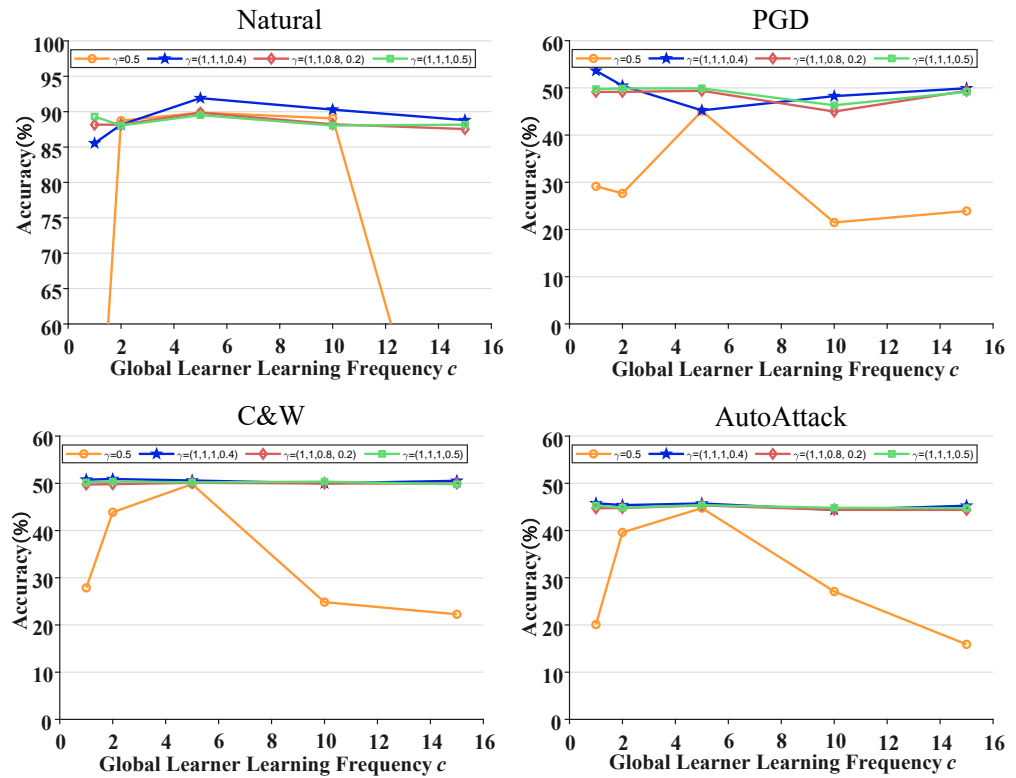


Figure 5. Comparison of different global learner learning frequencies c and two basic learner mixing ratios γ under the A3GT method. The experiments evaluated clean accuracy and robust accuracy against PGD, C&W, and AutoAttack attacks using ResNet-18 as the base model.

4.5.2. Learning Frequency c of the Global Learner

As shown in Equation (8), the learning frequency c of the global learner controls the frequency of interaction between the global learner and the two basic learners. A larger learning frequency c means lower interaction frequency, while a smaller c means more frequent interactions. It is worth noting that the learning frequency c remains fixed throughout the entire model training phase. The experimental results in Figure 5 indicate that the method is not optimal when $c = 1$. The performance reaches its peak when the learning frequency increases to $c = 5$. As the learning frequency continues to increase, the performance gradually decreases. This suggests that the interaction frequency between the global learner and the basic learners should neither be too high nor too low. Therefore, the A3GT method sets the learning frequency c of the global learner to 5.

5. Conclusions

This paper proposes an Asynchronous Adaptive Adversarial Training (A3GT) method to enhance model adversarial robustness. Following the training framework of Generalist, A3GT divides model training into two parts: normal training and adversarial training. The learner for normal training is called the clean learner, and the learner for adversarial

training is called the robust learner. The two basic learners operate independently and do not interfere with each other in the initial stages. To eliminate reliance on preset hyperparameters, A3GT uses an adaptive global learner update strategy, where the two learners interact to generate the global learner once both reach a stable state. By determining whether the two basic learners have stabilized, A3GT creates an adaptive interaction strategy based on different model architectures and datasets without being constrained by fixed static parameters.

The performance of A3GT is compared with other defense methods under different adversarial attacks and on different datasets. Experimental results show that A3GT not only exhibits good adversarial robustness under various attacks but also maintains classification accuracy on clean samples. In addition to using classification accuracy to evaluate performance, this paper also embeds A3GT into the Comprehensive Multi-dimensional Model Robustness (CMMR) evaluation framework. Evaluations with other advanced defense methods show that A3GT demonstrates good and stable performance across the entire range of adversarial perturbation strengths.

Although A3GT effectively balances adversarial robustness and classification accuracy, and demonstrates superior robustness under various adversarial attacks, it increases the training overhead to some extent due to the need to train two independent learners. However, since the two basic learners are trained independently, the learning speed can be accelerated by enhancing parallelism when sufficient computational resources are available. Additionally, this paper focuses on adversarial attacks in the image domain, and we believe that extending A3GT to other domains (such as natural language processing or speech recognition) is an interesting direction for future research.

Author Contributions: Conceptualization, S.Z.; Methodology, Z.H. (Zeyi He) and W.L.; Resources, S.Z.; Data curation, Z.H. (Zheng Huang) and Y.C.; Writing—original draft, Z.H. (Zheng Huang) and Y.C.; Writing—review & editing, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China grant number 62372473 and the Hunan Province Natural Science Foundation of China grant number 2023JJ70016.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: Authors Zeyi He and Zheng Huang were employed by the company Changsha Urban Development Group Co., Ltd. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Kalischek, N.; Wegner, J.D.; Schindler, K. In the light of feature distributions: Moment matching for neural style transfer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 9382–9391.
2. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
3. Wang, Z.; Wang, X.; Ma, J.; Qin, Z.; Ren, J.; Ren, K. Survey on Adversarial Example Attack for Computer Vision Systems. *Chin. J. Comput.* **2023**, *46*, 436–468.
4. Zhao, S.; Ma, B. MossFormer: Pushing the Performance Limit of Monaural Speech Separation Using Gated Single-Head Transformer with Convolution-Augmented Joint Self-Attentions. *arXiv* **2023**, arXiv:2302.11824.
5. Gui, T.; Xi, Z.; Zheng, R.; Liu, Q.; Ma, R.; Wu, T.; Bao, R.; Zhang, Q. Recent Researches of Robustness in Natural Language Processing Based on Deep Neural Network. *Chin. J. Comput.* **2024**, *47*, 90–112.
6. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
7. Ma, C.; Shen, C.; Lin, C.; Li, Q.; Wang, Q.; Li, Q.; Guan, X. Attacks and Defenses for Autonomous Driving Intelligence Models. *Chin. J. Comput.* **2024**, *1*, 1–22.

8. Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. Planning-oriented autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 17853–17862.
9. Sha, H.; Mu, Y.; Jiang, Y.; Chen, L.; Xu, C.; Luo, P.; Li, S.E.; Tomizuka, M.; Zhan, W.; Ding, M. LanguageMPC: Large Language Models as Decision Makers for Autonomous Driving. *arXiv* **2023**, arXiv:cs.RO/2310.03026.
10. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
11. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
12. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 39–57.
13. Wang, Z.; Guo, H.; Zhang, Z.; Liu, W.; Qin, Z.; Ren, K. Feature importance-aware transferable adversarial attacks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 7639–7648.
14. Zhong, Y.; Liu, X.; Zhai, D.; Jiang, J.; Ji, X. Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 15345–15354.
15. 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, Stockholm, Sweden, 10–15 July 2018; PMLR: London, UK, 2018; pp. 274–283.
16. Wang, H.; Wang, Y. Generalist: Decoupling natural and robust generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 20554–20563.
17. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2017**, arXiv:1706.06083.
18. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019; PMLR: London, UK, 2019; pp. 7472–7482.
19. Carmon, Y.; Ragunathan, A.; Schmidt, L.; Duchi, J.C.; Liang, P.S. Unlabeled data improves adversarial robustness. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 11190–11201.
20. Najafi, A.; Maeda, S.I.; Koyama, M.; Miyato, T. Robustness to adversarial perturbations in learning from incomplete data. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 5542–5552.
21. Lee, S.; Lee, H.; Yoon, S. Adversarial vertex mixup: Toward better adversarially robust generalization. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 272–281.
22. Zhang, J.; Xu, X.; Han, B.; Niu, G.; Cui, L.; Sugiyama, M.; Kankanhalli, M. Attacks which do not kill training make adversarial learning stronger. In Proceedings of the International Conference on Machine Learning, Online, 13–18 July 2020; PMLR: London, UK, 2020; pp. 11278–11287.
23. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the Artificial Intelligence and Statistics, Lauderdale, FL, USA, 20–22 April 2017; PMLR: London, UK, 2017; pp. 1273–1282.
24. Lamb, A.; Verma, V.; Kannala, J.; Bengio, Y. Interpolated adversarial training: Achieving robust neural networks without sacrificing too much accuracy. In Proceedings of the 12th ACM Workshop on Artificial Intelligence and Security, London, UK, 15 November 2019; pp. 95–103.
25. Liu, Y.; Cheng, Y.; Gao, L.; Liu, X.; Zhang, Q.; Song, J. Practical evaluation of adversarial robustness via adaptive auto attack. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 15105–15114.
26. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 9185–9193.
27. Liu, W.; Zhang, S.; Wang, W.; Zhang, J.; Liu, X. CMMR: A Composite Multidimensional Models Robustness Evaluation Framework for Deep Learning. In *Algorithms and Architectures for Parallel Processing, Proceedings of the 23rd International Conference, ICA3PP 2023, Tianjin, China, 20–22 October 2023, Proceedings, Part V*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 238–256.

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.