软件学报 ISSN 1000-9825, CODEN RUXUEW
*Journal of Software*, [doi: 10.13328/j.cnki.jos.007528]
©中国科学院软件研究所版权所有.
E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

# 基于动量加速和任务均衡的目标检测对抗训练方法 *

陇 盛[1], 林 晨[2], 陶 蔚[1,3], 张 军[1], 陶 卿[4]

[1](国防科技大学 大数据与决策实验室, 湖南 长沙 410073)

[2](西南财经大学 数据科学与商业智能联合实验室, 四川 成都 610074)

[3](军事科学院 战略评估咨询中心, 北京 100091)

[4](合肥理工学院, 安徽 合肥 238076)

通讯作者: 陶卿, E-mail: taoqing@gmail.com

**摘 要**: 对抗训练作为提升深度神经网络对抗鲁棒性的核心策略,在图像分类任务中已得到广泛关注,但在目标检测领域中的研究较为匮乏.传统对抗训练通常依赖投影梯度下降法(Projected Gradient Descent,PGD)开展模型的鲁棒优化,然而对抗样本的迭代大幅延长了模型训练周期,成为限制对抗训练在目标检测这类计算密集型任务中实际部署的主要瓶颈.针对这个问题,本文提出了一种基于 Nesterov 加速梯度(Nesterov's Accelerated Gradient,NAG)的对抗训练方法,通过引入 NAG 动量机制加速算法收敛,该方法在得到与 PGD 所训练模型精度相当的同时,显著加快了对抗训练速度.此外,目标检测与图像分类最主要的区别在于目标边界框定位.然而我们观察到现有方法仍侧重于学习基于分类损失产生的对抗样本,忽视了定位在目标检测中的特殊性.本文设计了一种自适应损失重加权策略,以均衡训练中不同任务所衍生对抗样本的数量占比,促进模型聚焦定位以增强鲁棒性.在 PASCAL VOC 和 MS COCO 两个公开目标检测数据集上与现有的先进目标检测对抗训练方法进行实验对比验证了所提方法的有效性.

**关键词**: 目标检测;对抗训练;动量优化;多任务损失

中文引用格式: 陇盛,林晨,陶蔚,张军,陶卿.基于动量加速和任务均衡的鲁棒目标检测对抗训练方法.软件学报. http://www.jos.org.cn/1000-9825/7528.htm

英文引用格式: Long S, Lin C, Tao W, Zhang J, Tao Q. Adversarial Training for Object Detection with Momentum Acceleration and Task Balancing. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7528.htm

## Adversarial Training for Object Detection with Momentum Acceleration and Task Balancing

LONG Sheng[1], LIN Chen[2], TAO Wei[1,3], ZHANG Jun[1], TAO Qing[4]

[1](Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China)

[2](Joint Laboratory of Data Science and Business Intelligence, Southwestern University of Finance and Economics, Chengdu 610074, China)

[3](Center for Strategic Assessment and Consulting, Academy of Military Sciences, Beijing 100091, China)

[4](Hefei Institute of Technology, Hefei 238076, China)

**Abstract**: Adversarial training, a key strategy for enhancing the adversarial robustness of DNNs, has been widely studied in image classification but lacks sufficient research in object detection. Traditional adversarial training often relies on PGD for model robust optimization. However, the iterative process of generating adversarial examples prolongs model training, becoming a major bottleneck for deploying adversarial training in computationally intensive tasks like object detection. To address this, we propose an adversarial training method based on NAG. By introducing the NAG momentum mechanism, it accelerates algorithm convergence. This method maintains detection accuracy comparable to PGD-trained models while significantly improving adversarial training efficiency. Additionally, since object detection mainly differs from image classification in object bounding box localization, we design an adaptive loss re-weighting

* 基金项目: 国家自然科学基金(62576351, 62076252, 62106281); 中国博士后科学基金第 76 批面上项目资助(2024M764294); 湖南省研究生科研创新项目(CX20240105)

收稿时间: 2025-05-12; 修改时间: 2025-06-30, 2025-08-15; 采用时间: 2025-08-20; jos 在线出版时间: 2025-09-02

strategy. It balances the number of adversarial examples from different tasks during training, restoring the model's focus on localization to enhance robustness. Experiments on the PASCAL VOC and MS COCO datasets, comparing our method with existing advanced object detection adversarial training approaches, validate the effectiveness of our proposed method.

**Key words**: object detection; adversarial training; momentum optimization; multi-task loss

在原始输入中添加人眼难以察觉的噪声,就能使基于神经网络的预测系统产生错误输出,这种精心设计的对抗样本最早在视觉分类任务中揭示了深度学习模型潜在的脆弱性[1][2][3],引发了人们对人工智能算法安全的担忧.为了提升模型抵御对抗攻击的鲁棒性能,研究者聚焦分类任务研究了多种对抗防御方法,例如对抗训练[4][5][6][7][8]、对抗净化[9][10]、对抗样本检测[11][12]和认证防御[13][14]等.其中最有效的防御手段是对抗训练,通过生成对抗样本加入训练集,让模型学习鲁棒特征,从而提高其在面对对抗攻击时的稳定性和可靠性.对抗训练作为一种提升模型鲁棒性的关键技术,已成为人工智能安全领域的研究热点.

目标检测作为计算机视觉的核心技术已经在自动驾驶[15]、医疗诊断[16]等现实场景中得到广泛应用,其展现出的强大检测性能对革新人类生产生活方式有着深远的影响,同时也为算法的安全鲁棒问题带来了更加严峻的挑战.Lu 等人发现对抗样本在检测任务中同样有效,并且成功欺骗了 Faster R-CNN 和 YOLO 两个经典目标检测模型[17].Xie 等人将对抗攻击扩展到目标检测和语义分割任务中,提出首个特定于稠密预测任务的对抗样本生成方法 DAG[18].随后涌现出了各种各样针对目标检测模型的对抗攻击算法,例如 UPC[19]、PRFA[20]、ZQA[21]、T-SEA[22]、基于 GAN 的攻击[23][24]等.对抗攻击算法的快速演变,也为防御手段的发展奠定了基础,虽然图像分类中已有诸多成熟的对抗样本防御方法,但目标检测领域的相关研究稍显滞后[25],且因其需同步完成分类与定位任务、网络结构复杂,致使图像分类防御方法难以直接套用,专用防御设计困难重重.

为了应对上述挑战,Zhang 等人从多任务学习的视角对目标检测模型训练过程展开分析,提出了多任务域对抗训练方法(Multi-Task-Domains, MTD),通过区别分类和回归损失生成多源对抗样本来实现模型鲁棒性的提升[26].在此基础上,Chen 等人进一步讨论了训练数据类均衡问题,通过设计全新的类别加权损失,提出基于类别的对抗训练(Class-Wise Adversarial Training, CWAT)方法,从而更加均匀、高效地强化了目标检测模型针对所有类别的对抗鲁棒性[27].尽管 MTD 和 CWAT 均在一定程度上有助于模型免疫对抗攻击,但却牺牲了正常输入样本的识别精度,因此,对抗鲁棒性和干净准确率之间的权衡同样存在于目标检测中并亟待解决.Dong 等人系统分析了模型在学习干净样本和对抗样本之间产生冲突的原因及其对分类和定位的影响,提出基于对抗感知卷积模块的目标检测模型 RobustDet 来缓解这个问题,并结合对抗图像鉴别器和一致性特征重构来进一步增强模型鲁棒性[28].RobustDet 已经成为目前最先进的基于对抗训练的目标检测器,然而,同传统方法一样,高昂的训练时间成本仍然是限制模型实际部署的主要瓶颈.

对抗训练通常被形式化描述为一个 min-max 鲁棒优化过程,无论是面对分类还是目标检测模型,与传统训练方式相比,在使用迭代次数为 $K$ 的 PGD 算法(记为 PGD-$K$)近似解内部最大化问题时,除了更新网络参数所需的梯度计算外,还需要 $K$ 步前向和反向传播来生成对抗样本.这意味着对抗训练的运行时间增加了 $K$+1 倍,采用如此耗时的训练过程处理大规模问题,对于计算密集的目标检测模型来说几乎是不可行的.换句话说,加速对抗训练的关键在于提高生成对抗样本的效率,对抗训练运行速度的快慢取决于 $K$ 值的大小,同时还需保证模型的干净和鲁棒准确率不受损失.基于这一分析,本文考虑优化领域的加速方法来改进 PGD 算法.

NAG 动量方法一直是优化领域中备受关注的一阶算法,针对一般的光滑凸函数和确定梯度,NAG 有严格的理论保证将 PGD 的收敛速率从 $O(1/K)$提升一个数量级达到 $O(1/K^2)$[29],其中 $K$ 是算法迭代次数.通俗来讲,NAG 能以更少的迭代次数达到和 PGD 相同的精度.这不仅是传统优化领域的一项里程碑式的突破,更为动量算法在深度学习中加速神经网络模型训练奠定了坚实的基础.可想而知,使用 NAG 替代对抗训练中普遍使用的 PGD 来节省训练时间开销是十分有前景的.自 1983 年诞生并一举打破传统一阶算法收敛速率上限以来[29],NAG 动量优化器及其变体持续在多个领域展现出强大的加速性能[30][31].尽管 NAG 在分类任务中衍生出具有强大黑盒迁移攻击性能的算法,例如 NI-FGSM[32]及其变体[33][34],但这些只利用了动量稳定更新方向的特性.目前还没有关于动量算法在缩短目标检测模型对抗训练时间方面的研究,因此本文认为 NAG 动量方法的巨大

加速潜力尚未在对抗攻防领域得到完全开发,并期望在目标检测对抗训练中,进一步探索优化理论中 NAG 加速算法收敛的现实意义.

此外,在目标检测与图像分类的比较中,关键区别在于目标边界框的精确定位.尽管文献[26]强调并分析了目标检测中不同任务损失之间的相互影响,提出多任务损失训练框架 MTD,从分类损失和定位损失中筛选攻击性更强的样本.但我们在实际应用中发现,无论是用定位损失攻击还是分类损失攻击生成对抗样本,最终用于模型训练时反馈的分类损失数值(loss_cls)总是大于定位损失(loss_loc),如图 1(a)和(b)所示.也就是说图像分类对抗样本在现有的目标检测对抗训练中仍然占据主导地位,如图 1(c)所示,MTD 偏向于选择分类损失衍生的对抗样本(x_cls)来进行训练,却忽略了定位在目标检测环节所具有的独特关键性.对定位对抗样本(x_loc)的边缘化破坏了多任务损失框架下选择对抗样本的公平性,导致了对抗训练中任务不均衡现象发生,进而可能阻碍目标检测模型的鲁棒性的提升.因此,重新审视定位损失在鲁棒目标检测模型训练过程中的作用,研究多任务均衡如何影响鲁棒性是一个值得探索的问题.


(a)定位攻击的损失曲线 (b)分类攻击的损失曲线 (c)分类和定位对抗样本数量

图 1 不同攻击下的模型损失比较以及不同源对抗样本数量

针对上述问题,本文主要贡献包括以下几个方面:

(1) 推导了 NAG 动量从基于梯度下降优化方法到其基于梯度上升对抗变体算法 $\mathrm{NAG}_{\mathrm{adv}}$ 的转换过程,因此建立了 NAG 在优化理论中加速收敛和实际应用中加速对抗样本生成之间的联系,并得到了凸和非凸情况下的与传统 NAG 匹配且优于 PGD 的收敛速率,为引入 $\mathrm{NAG}_{\mathrm{adv}}$ 替换 PGD 来提高目标检测模型对抗训练效率提供了理论支撑.

(2) 基于分类损失通常主导梯度变化的发现,本文提出了一个全新的自适应加权多任务损失函数,通过在训练过程中均衡不同任务所衍生的对抗样本的数量,加强目标检测对抗训练对定位这一特定任务的关注,以提升模型整体鲁棒性.

(3) 在 PASCAL VOC 和 MS COCO 两个公开目标检测数据集上的实验验证了本文方法的有效性,相同条件下与基于 PGD 的标准对抗训练模型相比,训练速度提升 1 倍.与当前先进目标检测对抗训练方法相比,本文方法鲁棒精度提升了 5.77%~11.74%.

## 1 相关工作

给定一个干净样本 $\boldsymbol{x}\in\mathbb{R}^d$,对抗攻击的目的是制造相应的对抗样本 $\bar{\boldsymbol{x}}$ 诱导参数为 $\boldsymbol{\theta}$ 的深度神经网络模型 $f_{\boldsymbol{\theta}}$ 预测错误.对抗训练抵御攻击的思路是:提前生成这种"有毒"的对抗样本并加入训练过程,刺激模型产生"抗体",从而达到鲁棒性提升的目的以免疫今后可能遭遇的对抗攻击.对抗训练方法最早在图像分类任务上出现,随后逐渐发展到目标检测中,下面分别从这两方面阐述相关工作.

### 1.1 图像分类对抗训练

Goodfellow 等人[3]最早提出利用 FGSM 生成对抗样本参与训练来提升模型鲁棒性,计算公式如下:

$$\bar{\boldsymbol{x}}=\boldsymbol{x}+\epsilon\cdot\mathrm{sign}\left(\nabla_{\boldsymbol{x}}\mathcal{L}\left(f_{\boldsymbol{\theta}}(\boldsymbol{x},y)\right)\right) \tag{1}$$

其中,$y$ 是 $\boldsymbol{x}$ 的真实标签,$\mathcal{L}$ 代表损失函数,$\epsilon$ 是扰动强度,$\epsilon$ 选取不当可能会导致模型过拟合于 FGSM 生成的对

抗样本.因此,Madry 等人[5]提出通过求解 min-max 鞍点问题来搜索最优的对抗样本和鲁棒模型:

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\max_{\bar{\boldsymbol{x}}\in\mathcal{S}_{\boldsymbol{x}}}\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}},y\right)\right)\right] \tag{2}$$

其中,$\mathcal{D}$ 表示训练数据分布,$\mathcal{S}_{\boldsymbol{x}}=\left\{\boldsymbol{z}\middle|\|\boldsymbol{z}-\boldsymbol{x}\|\le\epsilon\right\}$.FGSM 可以看作是对内部最大化问题的一个近似求解,更好的方法是在内层采用多步的 PGD,为外层最小化损失优化模型提供数据支撑.PGD 迭代公式如下:

$$\bar{\boldsymbol{x}}_0=\boldsymbol{x},\quad \bar{\boldsymbol{x}}_{k+1}=\mathcal{P}_{\mathcal{S}_{\boldsymbol{x}}}\left(\bar{\boldsymbol{x}}_k+\epsilon\cdot\operatorname{sign}\left(\nabla_{\boldsymbol{x}}\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k,y\right)\right)\right)\right) \tag{3}$$

其中,$\mathcal{P}_{\mathcal{S}_{\boldsymbol{x}}}$ 表示投影算子,将输入投影到可行域 $\mathcal{S}_{\boldsymbol{x}}$.

尽管基于 PGD 的对抗训练方法能够在很大程度上提升模型的鲁棒性,但是不可避免地带来了巨额的计算开销.基于 PGD-$K$ 的标准对抗训练方法每次模型迭代所需梯度计算次数为 $O(M(K+1))$,$M$ 是样本数量.而传统训练方法仅需 $O(M)$,也就是说对抗训练方法比传统训练方法慢 $K+1$ 倍.为解决计算效率的问题,Shafahi 等人[35]提出"免费"对抗训练,通过循环利用更新模型参数时计算的梯度信息来消除生成对抗样本的开销,即在一次反向传播中同时计算损失相对于模型参数和输入图像的梯度,"免费"对抗训练每次模型迭代所需梯度计算次数为 $O(MN)$,为了抵消小批量循环的额外计算成本,将迭代总数减少了 $N$ 倍,因此实际训练总时长远小于 $T$ 次迭代的标准对抗训练.尽管这样的方式减少了计算量,但在实际中仍然比同等轮次的传统训练方式慢.为此 Wong 等人[36]重新审视了基于 FGSM 的对抗训练鲁棒性不足的原因,提出结合随机初始化和 FGSM 的快速对抗训练方法,该方法在得到与"免费"对抗训练方法同等精度模型的情况下,将训练时长从 10 小时缩短到了 6 分钟,证明了从节省内层最大化问题计算开销的角度来加速对抗训练成效显著.随后也衍生出一些新的快速对抗训练方法,以进一步克服 FGSM 面临的灾难性过拟合问题[37][38].

### 1.2 目标检测对抗训练

区别于图像分类,目标检测模型需要输出干净图像 $\boldsymbol{x}$ 中目标的边界框 $\hat{\boldsymbol{b}}_i=\left[p_i^x,p_i^y,w_i,h_i\right]$ 及其预测类别概率 $\hat{\boldsymbol{c}}_i=\left[\hat{c}_i^{bg},\hat{c}_i^1,...,\hat{c}_i^C\right]$,其中 $p_i^x$ 和 $p_i^y$ 是第 $i$ 个目标的左上角坐标值,$w_i$ 和 $h_i$ 是第 $i$ 个目标的宽和高,$C$ 为类别总数,$bg$ 代表背景.Zhang 等人[26]最早将目标检测对抗训练形式化描述为如下鲁棒优化问题:

$$\min_{\boldsymbol{\theta}} \ \mathbb{E}_{(\boldsymbol{x},\{\boldsymbol{b}_i,\boldsymbol{c}_i\})\sim\mathcal{D}}\left[\max_{\bar{\boldsymbol{x}}\in\mathcal{S}_{\text{loc}}\cup\mathcal{S}_{\text{cls}}}\mathcal{L}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}},y\right)\right)\right] \tag{4}$$

其中,$y=\left\{\boldsymbol{b}_i,\boldsymbol{c}_i\right\}$ 代表 $\boldsymbol{x}$ 中第 $i$ 个目标的真实边界框和类别,总损失由定位任务损失(通常为 smooth $L_1$ 损失)和分类任务损失(通常为交叉熵损失)组成,即 $\mathcal{L}=\mathcal{L}_{\text{loc}}+\mathcal{L}_{\text{cls}}$.$\mathcal{S}_{\text{loc}}$ 和 $\mathcal{S}_{\text{cls}}$ 分别表示在这两个损失上生成对抗样本的任务域:

$$\mathcal{S}_{\text{loc}}\triangleq\left\{\bar{\boldsymbol{x}}\middle|\arg\max_{\bar{\boldsymbol{x}}\in\mathcal{S}_{\boldsymbol{x}}}\mathcal{L}_{\text{loc}}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}},\{\boldsymbol{b}_i\}\right)\right)\right\},\quad \mathcal{S}_{\text{cls}}\triangleq\left\{\bar{\boldsymbol{x}}\middle|\arg\max_{\bar{\boldsymbol{x}}\in\mathcal{S}_{\boldsymbol{x}}}\mathcal{L}_{\text{cls}}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}},\{\boldsymbol{c}_i\}\right)\right)\right\} \tag{5}$$

直接将图像分类对抗训练方法迁移到目标检测,等价于在任务无关的域 $\mathcal{S}_{\boldsymbol{x}}$ 上训练模型,实践证明这种训练方式会因为不同任务之间存在的冲突而导致模型鲁棒性受限.因此提出基于多任务域 $\mathcal{S}_{\text{loc}}\cup\mathcal{S}_{\text{cls}}$ 的对抗训练方法 MTD,在避免任务之间的互相干扰的同时,分别从定位和分类损失引导的对抗样本中获取最大收益.为了提高计算效率,MTD 采用 FGSM 近似求解符合任务域对抗样本,在此基础上,Chen 等人[27]进一步考虑了给定图像中某一特定类别的目标数量大于其他类别的情况,提出了类加权的对抗训练方法 CWAT.与以往的防御方法相比,CWAT 不仅能够平衡各类的影响,而且能够有效、均匀地提高训练模型对所有目标类的对抗鲁棒性.然而这些方法都存在一个共性问题——目标检测模型对干净样本的准确性和对抗样本的鲁棒性之间的权衡.为了缓解这个问题,Dong 等人[28]提出了基于对抗感知卷积的 RobustDet 模型,利用对抗图像鉴别器(Adversarial Image Discriminator, AID)为干净样本和对抗样本生成不同的权重,从而引导对抗感知卷积核自适应地学习鲁棒特征.RobustDet 还利用一致性特征重建(Consistent Features with Reconstruction, CFR)将对抗图像重建为干净样本,以进一步增强鲁棒性.RobustDet 使用"免费"对抗训练模式,在训练速度、干净样本准确性和对抗样本鲁棒性几项指标上,已经成为目前最先进的基于对抗训练的鲁棒目标检测器.

## 2 动量加速和任务均衡的目标检测对抗训练方法

### 2.1 基于NAG动量的对抗训练速度提升

受到快速对抗训练方法在图像分类任务上加速效果显著的启发,本文通过节省鲁棒优化问题中内层最大化过程的开销,以提升目标检测模型对抗训练速度,同时为了避免 FGSM 算法容易导致的灾难性过拟合问题,我们不是极端地选择单步对抗样本生成方式,而是探索能以更少迭代实现与 PGD-$K$ 相同或更高鲁棒精度的算法.因此,我们聚焦于收敛速率具有数量级提升的 NAG[29][39],其关键迭代步骤如下:

$$\begin{cases}\boldsymbol{\theta}_k=\boldsymbol{\vartheta}_k-\eta_k F'(\boldsymbol{\vartheta}_k)\\ a_{k+1}=\dfrac{1}{2}\left(1+\sqrt{4a_k^2+1}\right)\\ \boldsymbol{\vartheta}_{k+1}=\boldsymbol{\theta}_k+(a_k-1)(\boldsymbol{\theta}_k-\boldsymbol{\theta}_{k-1})/a_{k+1}\end{cases} \tag{6}$$

其中,$k\in[0,K]$,$a_0=1$,$\boldsymbol{\theta}_{-1}=\boldsymbol{\vartheta}_0$,$\boldsymbol{\vartheta}_0$ 为模型初始权重,$F(\boldsymbol{\theta})=\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x},y))$,$F'(\boldsymbol{\theta})=\nabla_{\boldsymbol{\theta}}\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x},y))$,$\eta_k=2^{-i}\eta_{k-1}$,$i$ 是使得 $F(\boldsymbol{\vartheta}_k)-F(\boldsymbol{\vartheta}_k-2^{-i}\eta_{k-1}F'(\boldsymbol{\vartheta}_k))\geq 2^{-i-1}\eta_{k-1}\|F'(\boldsymbol{\vartheta}_k)\|^2$ 成立的最小正整数,且对任意 $z\neq\boldsymbol{\vartheta}_0$,$\eta_{-1}=\|\boldsymbol{\vartheta}_0-z\|/\|F'(\boldsymbol{\vartheta}_0)-F'(z)\|$.NAG 在光滑凸情形下能达到一阶算法的最优收敛速度 $O(1/K^2)$,如引理 1 所示.

**假设 1.** $F$ 是光滑目标函数,即 $\exists L_F>0$,使得

$$\|F'(\boldsymbol{a})-F'(\boldsymbol{b})\|\leq L_F\|\boldsymbol{a}-\boldsymbol{b}\|,\qquad \forall \boldsymbol{a},\boldsymbol{b}\in\mathbb{R}^d$$

**引理 1**(文献[29]定理 1)**.** 令假设 1 成立,$\{\boldsymbol{\theta}_k\}_{k=0}^K$ 由(6)式产生,$\boldsymbol{\theta}^*$ 为 $F(\boldsymbol{\theta})$ 最优解.若 $F$ 为凸函数,那么下式成立:

$$F(\boldsymbol{\theta}_K)-F(\boldsymbol{\theta}^*)\leq\frac{4L_F\|\boldsymbol{\vartheta}_0-\boldsymbol{\theta}^*\|^2}{(K+2)^2} \tag{7}$$

在 NAG 基础上,本文初始化干净样本 $\boldsymbol{x}$ 作为优化对象,提出了 NAG 在对抗训练中生成对抗样本 $\bar{\boldsymbol{x}}_k$ 的变体算法 $\text{NAG}_{\text{adv}}$:

$$\begin{cases}\bar{\boldsymbol{x}}_k=\arg\min\limits_{z\in\mathcal{S}_x}\left\{F(\boldsymbol{x}_k)+\langle F'(\boldsymbol{x}_k),z-\boldsymbol{x}_k\rangle+\frac{1}{2\eta_k}\|z-\boldsymbol{x}_k\|^2\right\}\\ a_{k+1}=\dfrac{1}{2}\left(1+\sqrt{4a_k^2+1}\right)\\ \boldsymbol{x}_{k+1}=\bar{\boldsymbol{x}}_k+(a_k-1)(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1})/a_{k+1}\end{cases}\qquad,\forall k\geq 0 \tag{8}$$

其中,$\bar{\boldsymbol{x}}_{-1}=\boldsymbol{x}_0=\boldsymbol{x}$,$F(\boldsymbol{x})=-\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}))$,$F'(\boldsymbol{x})=-\nabla_x\mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}))$.

根据引理 1,不难发现(8)式 $\text{NAG}_{\text{adv}}$ 能以 $O(1/K^2)$的速率在光滑凸环境中收敛到(4)式最大化问题的最优点,相比之下 PGD 较慢只能达到 $O(1/K)$.为进一步分析非凸环境收敛性以及便于算法实现,我们给出了 $\text{NAG}_{\text{adv}}$ 的两种等价形式,如定理 1 和定理 2 所示.

**定理 1.** 令 $\sigma_{k+1}\equiv 1/a_{k+1}$,$z_k\equiv\sigma_k^{-1}(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}(1-\sigma_k))$,$\mathcal{S}_{\boldsymbol{x}}=\mathbb{R}^d$,$\text{NAG}_{\text{adv}}$ 可等价转换为下式:

$$\begin{cases}\boldsymbol{x}_k=(1-\sigma_k)\bar{\boldsymbol{x}}_{k-1}+\sigma_k z_{k-1}\\ z_k=z_{k-1}-\sigma_k^{-1}\eta_k F'(\boldsymbol{x}_k)\\ \bar{\boldsymbol{x}}_k=(1-\sigma_k)\bar{\boldsymbol{x}}_{k-1}+\sigma_k z_k\end{cases}\qquad,\forall k\geq 1 \tag{9}$$

证明见附录 A.

由于(9)式与文献[42]中算法 1 等价,我们介绍如下引理进一步证明收敛性.

**引理 2**(文献[42]推论 1a)**.** 令假设 1 成立,$\boldsymbol{x}^*$ 为 $F(\boldsymbol{x})$ 最优解.设 $\alpha_k=2/k+1$,$\beta_k\equiv 1/2L_F$,若对 $\forall k\geq 1$ 满足 $\lambda_k\in[\beta_k,(1+(\alpha_k/4))\beta_k]$,那么下式成立:

$$\min_{k=1,\ldots,K}\left\|F'\left(\bar{\boldsymbol{x}}_k\right)\right\|^2 \le \frac{6L_F\left[F\left(\boldsymbol{x}\right)-F\left(\boldsymbol{x}^*\right)\right]}{K} \tag{10}$$

由于 $\eta_k \le 1/2L_F$ , $\sigma_k \le 2/k+1$ 在(9)式中恒成立,令 $\lambda_k \equiv \sigma_k^{-1}\eta_k$ , $\beta_k \equiv \eta_k$ , $\alpha_k \equiv \sigma_k$ ,不难发现(9)式同样满足引理 2 条件,因此 $\text{NAG}_{\text{adv}}$ 能以 $O(1/K)$的速率在非凸环境中收敛到(4)式最大化问题的一阶稳定点.

**定理 2.** 令 $\mu_k \equiv \eta_k\eta_{k+1}^{-1}\left(a_k-1\right)/a_{k+1}$ , $\mathcal{S}_{\boldsymbol{x}}=\mathbb{R}^d$ ,$\text{NAG}_{\text{adv}}$ 可等价转变为下式:

$$\begin{cases}\bar{\boldsymbol{x}}_k^{\text{nes}} \equiv \bar{\boldsymbol{x}}_k + \eta_{k+1}\mu_k\boldsymbol{g}_k \\ \boldsymbol{g}_{k+1} = \mu_k\boldsymbol{g}_k - F'\left(\bar{\boldsymbol{x}}_k^{\text{nes}}\right) \\ \bar{\boldsymbol{x}}_{k+1} = \bar{\boldsymbol{x}}_k + \eta_{k+1}\boldsymbol{g}_{k+1}\end{cases} \quad ,\forall k \ge 0 \tag{11}$$

证明见附录 B.

注意,(11)式与 NI-FGSM[32]近似等价,很多研究者认为可以使用 NI-FGSM 替代 PGD 实现加速,但是本文的 $\text{NAG}_{\text{adv}}$ 方法与之不同.区别在于 NI-FGSM 额外对 $\boldsymbol{g}_k$ 进行 $\text{L}_1$ 范数归一化和符号函数取值,这一系列操作有利于迁移对抗攻击性能的提升,但失去了收敛性保证.相反,本文方法通过与原始 NAG 的动量系数 $\mu_k$ 和步长系数 $\eta_{k+1}$ 对齐,建立起优化理论收敛速率与对抗样本生成速度之间的联系,从理论上证明了 $\text{NAG}_{\text{adv}}$ 具有比 PGD 更好的加速效果.

### 2.2 基于任务均衡的对抗鲁棒性提升

分类损失和定位损失在目标检测任务中通常有不同的量纲和变化范围.分类损失通常与概率分布相关,在与真实概率差异明显的训练初期,数值可能会比较大,此外分类任务也可能因关注难负样本的优化而产生较大的损失数值.而定位损失通常基于交并比(Intersection of Union, IoU)或几何距离组成的 smooth $L_1$ 、均方误差等,它们对误差的度量相对较为温和,因此值域可能更小.

我们从图 1(a)和(b)中也观察到了这一现象,分类损失值总是高于定位损失值.这种差异会导致两者对总损失的贡献不均衡,影响模型对抗训练的平衡性.尽管文献[27]考虑了任务之间可能存在的不平衡现象,但只是通过简单地设置阈值对超出部分损失值进行截断,以防止某一任务损失无限增加主导总损失.然而这样的策略需要额外引入一个手工设置的超参数,并针对不同模型重新调整优化,增加了调参负担.

我们提出了一种动态损失重加权(Loss Reweighting, LR)方案缓解上述问题,具体来说,设计了一个自适应损失权重参数,该参数由当前分类和定位损失在总损失中的比重决定,无需人工调参,计算方式如下:

$$\begin{aligned}\text{avg}_{\text{loc}} &= \frac{1}{N_o}\sum_{i=0}^{N_o}\mathcal{L}_{\text{loc}}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k,\{\boldsymbol{b}_i\}\right)\right), \\ \text{avg}_{\text{cls}} &= \frac{1}{N_o}\sum_{i=0}^{N_o}\mathcal{L}_{\text{cls}}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k,\{\boldsymbol{c}_i\}\right)\right), \\ \alpha &= \text{avg}_{\text{loc}}/\left(\text{avg}_{\text{loc}}+\text{avg}_{\text{cls}}+1\text{e}-5\right)\end{aligned} \tag{12}$$

其中, $N_o$ 为干净样本 $\boldsymbol{x}$ 所含的目标数量,上式用平均值 $\text{avg}_{\text{loc}}$ 和 $\text{avg}_{\text{cls}}$ 衡量定位损失和分类损失的相对大小,从而自适应地动态调整权重 $\alpha$ ,然后对定位和分类损失进行重加权分别得到 $\mathcal{L}_{\text{locLR}}$ 和 $\mathcal{L}_{\text{clsLR}}$ ,最后以凸组合形式求和得到任务均衡损失 $\mathcal{L}_{\text{LR}}$ ,具体如下:

$$\begin{aligned}\mathcal{L}_{\text{locLR}} &= \left(1-\alpha\right)\cdot\mathcal{L}_{\text{loc}}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k,y\right)\right), \\ \mathcal{L}_{\text{clsLR}} &= \alpha\cdot\mathcal{L}_{\text{cls}}\left(f_{\boldsymbol{\theta}}\left(\bar{\boldsymbol{x}}_k,y\right)\right), \\ \mathcal{L}_{\text{LR}} &= \mathcal{L}_{\text{locLR}}+\mathcal{L}_{\text{clsLR}}\end{aligned} \tag{13}$$

通过上述方式,算法能够更公平地选择对鲁棒性贡献最大的任务域对抗样本,如图 1(c)所示,在未经过损失重加权的 MTD 中,基于分类和定位损失对抗样本的比例大致为 3:1,在加入我们的自适应损失重加权策略后比例约为 1:1,使得定位和分类任务在对抗训练中的贡献达到均衡,从而实现鲁棒性的提升.

### 2.3 基于NAG动量和自适应损失重加权的目标检测对抗训练方法

结合上述的 NAG 动量对抗样本生成方法和损失重加权策略,本文提出了一个全新的目标检测对抗训练方法 $\text{NAG}_{\text{adv}}$-LR-$K$,详细流程见算法 1.

算法 1. 基于 NAG 动量和自适应损失重加权的目标检测对抗训练方法($\text{NAG}_{\text{adv}}$-LR-$K$)

输入:训练样本 $\{(\boldsymbol{x},\{\boldsymbol{b}_i,\boldsymbol{c}_i\})_m\}_{m=0}^{M-1}\sim\mathcal{D}$,模型训练次数 $T$、学习率 $\eta$,扰动半径 $\epsilon$.

for $t=0$ to $T-1$ do

  for $m=0$ to $M-1$ do

    根据(12)式计算自适应权重 $\alpha$

    for $k=0$ to $K-1$ do

      令 $F(\boldsymbol{x}_k)=-\mathcal{L}_{\text{locLR}}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k,\{\boldsymbol{b}_i\}\right)\right)$,根据(8)式计算 $\overline{\boldsymbol{x}}_{k+1}^{\text{loc}}$

      令 $F(\boldsymbol{x}_k)=-\mathcal{L}_{\text{clsLR}}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k,\{\boldsymbol{c}_i\}\right)\right)$,根据(8)式计算 $\overline{\boldsymbol{x}}_{k+1}^{\text{cls}}$

    end for

    $\gamma=\mathcal{L}_{\text{LR}}\left(f_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}}_K^{\text{loc}},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}\right)\right)>\mathcal{L}_{\text{LR}}\left(f_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}}_K^{\text{cls}},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}\right)\right)$

    $\overline{\boldsymbol{x}}=\gamma\overline{\boldsymbol{x}}_K^{\text{loc}}+(1-\gamma)\overline{\boldsymbol{x}}_K^{\text{cls}}$

    $\boldsymbol{\theta}=\boldsymbol{\theta}-\eta\cdot\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{LR}}\left(f_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}\right)\right)$

  end for

end for

输出: $\boldsymbol{\theta}$

算法 1 为本文方法在标准对抗训练模式下的实现,其单次模型迭代所需的梯度计算量为 $O(MK)$.为了进一步比较算法时效性,我们还给出了本文方法在"免费"对抗训练模式下的版本 NAG-LR-free,具体流程如算法 2 所示.NAG-LR-free 在单次模型迭代中需要梯度计算量同样为 $O(MK)$,但是模型迭代总数是 $\text{NAG}_{\text{adv}}$-LR-$K$ 的 $1/K$,因此实际训练效率会比 $\text{NAG}_{\text{adv}}$-LR-$K$ 更高.

算法 2. 基于 NAG 动量和自适应损失重加权的目标检测"免费"对抗训练方法($\text{NAG}_{\text{adv}}$-LR-free)

输入:训练样本 $\{(\boldsymbol{x},\{\boldsymbol{b}_i,\boldsymbol{c}_i\})_m\}_{m=0}^{M-1}\sim\mathcal{D}$,模型训练次数 $T$、学习率 $\eta$,扰动半径 $\epsilon$.

for $t=0$ to $(T-1)/K$ do

  for $m=0$ to $M-1$ do

    根据(12)式计算自适应权重 $\alpha$

    for $k=0$ to $K-1$ do

      令 $F(\boldsymbol{x}_k)=-\mathcal{L}_{\text{locLR}}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k,\{\boldsymbol{b}_i\}\right)\right)$,根据(8)式计算 $\overline{\boldsymbol{x}}_{k+1}^{\text{loc}}$

      令 $F(\boldsymbol{x}_k)=-\mathcal{L}_{\text{clsLR}}\left(f_{\boldsymbol{\theta}}\left(\boldsymbol{x}_k,\{\boldsymbol{c}_i\}\right)\right)$,根据(8)式计算 $\overline{\boldsymbol{x}}_{k+1}^{\text{cls}}$

      $\gamma=\mathcal{L}_{\text{LR}}\left(f_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}}_{k+1}^{\text{loc}},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}\right)\right)>\mathcal{L}_{\text{LR}}\left(f_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}}_{k+1}^{\text{cls}},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}\right)\right)$

      $\overline{\boldsymbol{x}}=\gamma\overline{\boldsymbol{x}}_{k+1}^{\text{loc}}+(1-\gamma)\overline{\boldsymbol{x}}_{k+1}^{\text{cls}}$

      $\boldsymbol{\theta}=\boldsymbol{\theta}-\eta\cdot\nabla_{\boldsymbol{\theta}}\mathcal{L}_{\text{LR}}\left(f_{\boldsymbol{\theta}}\left(\overline{\boldsymbol{x}},\{\boldsymbol{b}_i,\boldsymbol{c}_i\}\right)\right)$

    end for

  end for

end for

输出: $\boldsymbol{\theta}$

## 3 数值实验

单阶段目标检测算法因计算高效、部署便捷而被广泛应用于实际生产生活中,鉴于此,本文实验在基于VGG16骨干网络的单阶段多框检测器(single-shot multi-box detector, SSD)上进行对抗鲁棒性验证.

### 3.1 实验数据集

本文在PASCAL VOC[40]和MS-COCO[41]两个公开目标检测数据集上进行实验,表1列出了数据集的详细信息.在PASCAL VOC上开展实验时,采用标准的"07+12"训练验证集(即VOC 2007 trainval和VOC 2012 trainval)训练模型,由于VOC 2012测试集缺少生成对抗样本所必需的标注文件,因此仅在VOC 2007测试集(VOC 2007 test)上进行鲁棒性评估.在MS-COCO上开展实验时,采用COCO 2017训练集(COCO 2017 train)训练模型,在COCO 2017验证集(COCO 2017 val)上进行评估.

**表1** 实验数据集

| 数据集名称 | 子集划分 | 样本数量 | 类别数量 |
|---|---|---|---|
| PASCAL VOC[40] | VOC 2007 trainval | 5011 | 20 |
| | VOC 2012 trainval | 11540 | |
| | VOC 2007 test | 4952 | |
| MS-COCO[41] | COCO 2017 train | 118287 | 80 |
| | COCO 2017 val | 5000 | |

### 3.2 评价指标和对比算法

本文主要采用目标检测模型常用的平均精度(mean average precision, mAP)作为评价指标,其中交并比阈值为0.5.为了评价模型对干净样本的识别精度,计算模型在干净测试数据上的mAP值,为了评估模型鲁棒性,计算模型在基于分类损失的攻击($\mathcal{A}_{\text{cls}}$)、基于定位损失的攻击($\mathcal{A}_{\text{loc}}$)和基于类别的攻击(Class-Wise Attack, CWA)三种不同的对抗攻击下的mAP值.此外还引入模型训练时间来评价目标检测模型的对抗训练速度.

对比算法包括干净训练数据集上进行传统训练的SSD、分类任务域$\mathcal{S}_{\text{cls}}$上进行对抗训练的SSD变体(SSD-AT-$\mathcal{A}_{\text{cls}}$)、定位任务域$\mathcal{S}_{\text{loc}}$上进行对抗训练的SSD变体(SSD-AT-$\mathcal{A}_{\text{loc}}$).同时采用了一系列先进的目标检测对抗训练方法进行对比,包括MTD[26]、CWAT[27]和RobustDet[28].本文所提方法NAG$_{\text{adv}}$-LR在"免费"和标准对抗训练模式下的版本分别为NAG$_{\text{adv}}$-LR-free和NAG$_{\text{adv}}$-LR-$K$,其中$K$是NAG$_{\text{adv}}$的迭代次数.

### 3.3 实验细节

本文实验方法与文献[28]保持一致,所有模型使用学习率$\eta$为0.001、动量系数为0.9、权重衰减率为0.0005的随机梯度下降(Stochastic Gradient Descent, SGD)算法进行训练,在VOC数据集上模型训练次数$T$=40000,在COCO数据集上模型训练次数$T$=120000,同时结合了早停和混合精度训练机制.训练样本的图像尺寸为$300\times300$,像素值范围为[0, 255],并根据数据集的均值进行偏移.训练样本的batch size设置为32,每批次训练前随机抽取其中16个干净样本生成对抗样本,再与对应干净样本拼接成一组数据送入模型训练,对抗攻击均设定为非定向模式并生成无穷范数扰动,扰动半径$\epsilon$设置为8.本文所有方法的实现基于NVIDIA L20 GPU、ubuntu18.04.6操作系统、python3.9、pytorch1.13.1和torchvision0.14.1等软硬件设施.

### 3.4 实验结果与分析

我们首先参照文献[28]的实验设计与现有先进的对抗训练方法进行对比,由于MTD采用快速对抗训练模式,CWAT和RobustDet采用"免费"对抗训练模式,为公平起见,我们用与上述算法梯度计算量同阶的NAG-LR-free进行比较.在PASCAL VOC 2007测试集上评估干净样本检测精度、鲁棒精度以及对应模型的训练时间,结果如表2所示.

**表 2** 在 PASCAL VOC 2007 测试集上对比算法的干净和鲁棒 mAP(%)及训练时间(min)

| 算法 | 干净样本 | $\mathcal{A}_{cls}$ | $\mathcal{A}_{loc}$ | CWA | 训练时间 |
|---|---|---|---|---|---|
| SSD | **77.5** | 1.8 | 4.5 | 1.2 | **736** |
| SSD-AT- $\mathcal{A}_{cls}$ [26] | 46.7 | 21.8 | 32.2 | - | - |
| SSD-AT- $\mathcal{A}_{loc}$ [26] | 51.9 | 23.7 | 26.5 | - | - |
| MTD[26] | 48.0 | 29.1 | 31.9 | 18.2 | - |
| CWAT[27] | 51.3 | 22.4 | 36.7 | 19.9 | - |
| RobustDet | 75.4 | 41.5 | 45.2 | 42.4 | 1462 |
| $NAG_{adv}$-LR-free | 75.39 | **47.27** | **56.94** | **48.81** | 1481 |

从表 2 中可以看出,本文所提的 $NAG_{adv}$-LR-free 几乎不损失干净准确率,且鲁棒性能都超过对比算法,比 RobustDet 提升了 5.77%~11.74%.训练时间方面,与对比算法保持相当的训练速度,较传统非鲁棒的 SSD 模型训练增加的时间在可接受范围内.在 COCO 数据集上的结果同样验证了本文方法的有效性,如表 3 所示.

**表 3** 在 MS-COCO 2017 验证集上对比算法的干净和鲁棒 mAP(%)及训练时间(min)

| 算法 | 干净样本 | $\mathcal{A}_{cls}$ | $\mathcal{A}_{loc}$ | CWA | 训练时间 |
|---|---|---|---|---|---|
| SSD | **42.0** | 0.4 | 1.8 | 0.1 | - |
| MTD[26] | 24.2 | 13.0 | 13.4 | 7.7 | - |
| CWAT[26] | 23.7 | 14.2 | 15.5 | 9.2 | - |
| RobustDet | 36.7 | **20.6** | 19.4 | **20.5** | 1458 |
| $NAG_{adv}$-LR-free | 40.3 | 17.1 | **23.6** | 18.3 | **488** |

为进一步验证本文所提算法的加速效果,我们基于目前最先进的 RobustDet 模型,分别在快速对抗训练(FGSM)、"免费"对抗训练(PGD-free、$NAG_{adv}$-LR-free)和标准对抗训练(PGD-20、$NAG_{adv}$-LR-10)三种模式下测试干净样本检测精度和鲁棒精度,由于时间成本过高,我们将标准对抗训练的总轮数 $T$ 缩短到 5000,其余仍保持不变.结果如表 4 所示.

**表 4** 在 PASCAL VOC 2007 测试集上不同训练模式的干净和鲁棒 mAP(%)及训练时间(min)

| 算法 | 干净样本 | $\mathcal{A}_{cls}$ | $\mathcal{A}_{loc}$ | CWA | 训练时间 |
|---|---|---|---|---|---|
| FGSM | 75.92 | 39.38 | 49.65 | 40.81 | 1459 |
| PGD-free | 75.4 | 41.5 | 45.2 | 42.4 | 1462 |
| PGD-20 | 76.39 | 30.37 | 38.75 | 31.88 | 2343 |
| $NAG_{adv}$-LR-free | 75.39 | **47.27** | **56.94** | **48.81** | 1481 |
| $NAG_{adv}$-LR-10 | **77.10** | 41.55 | 55.27 | 42.99 | **1189** |

对比表 4 中训练轮次相同的 PGD-20 和 $NAG_{adv}$-LR-10 发现,$NAG_{adv}$-LR-10 不仅显著提升了鲁棒精度,比 PGD-20 高 11.11%~16.52%,还将时训练时间从 2343 分钟降到了 1189 分钟,速度提升 1 倍,这验证了前文所述通过减少内部迭代次数能够有效提高训练效率.对比训练轮次相同且梯度计算量同阶的 FGSM、PGD-free 和 NAGadv-LR-free 不难发现,尽管 FGSM 代表的快速对抗训练耗时最短,但由于缺少对同一批次数据多步优化的过程,导致平均性能比 PGD"免费"对抗训练略差,而本文的 NAGadv-LR-free 在保证相当训练速度和干净样本精度的同时,能够达到最高的鲁棒精度.

我们进一步在干净样本、基于分类攻击的对抗样本、基于定位攻击的对抗样本上通过可视化的方式对比了非鲁棒的 SSD(SSD)、基于 PGD-20 对抗训练的 RobustDet(PGD)和基于 $NAG_{adv}$-LR-10 对抗训练的 RobustDet(Ours)三个模型的检测结果,如图 2 所示.可以看出,三个模型均能在干净样本上准确识别出所有的目标类别和边界框,但是在对抗样本上的表现差异较大,其中,非鲁棒的 SSD 出现了严重的定位框消失和分类错误等问题.基于 PGD-20 对抗训练的 RobustDet 检测结果得到了明显改善,但是仍识别出了标签外的错误目标,且伴随着少量目标丢失的情况.相比之下,本文所提方法能够克服这些问题,无论是在干净样本还是对抗样本上都达到了最好的检测效果.

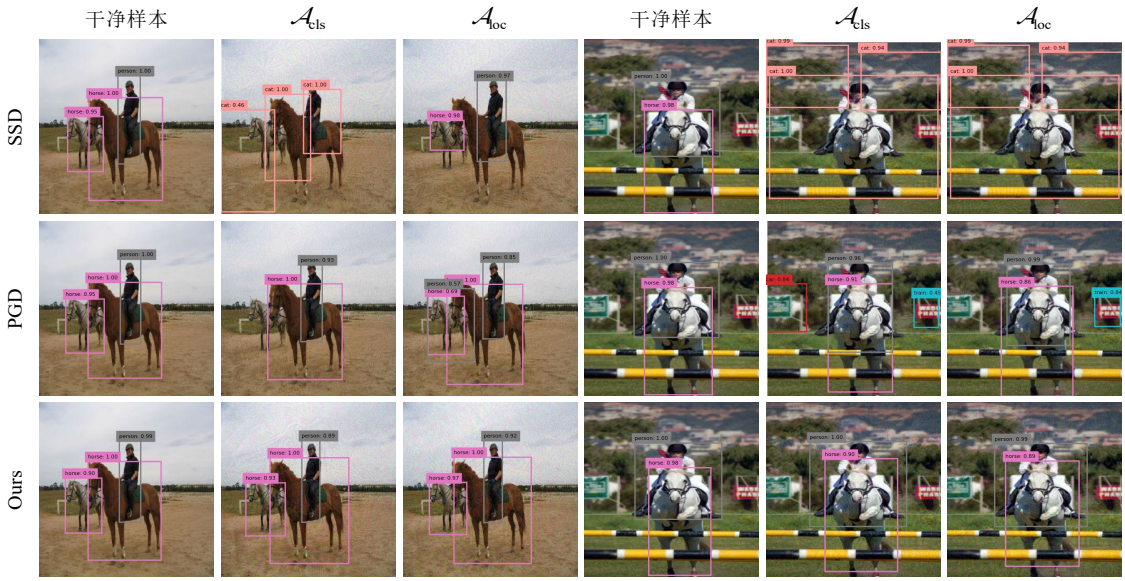



图 2 SSD、RobustDet 和本文方法所训练模型的检测结果

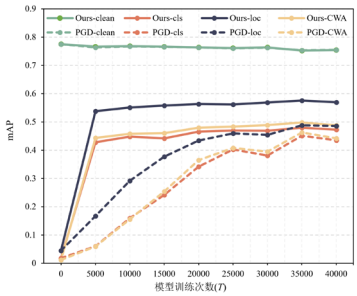
### 3.5 消融实验

本节分别研究了 NAG 动量方法($NAG_{adv}$)和损失重加权策略(LR)对整体性能的影响.首先,为提高效率,在"免费"对抗训练模式下进行评估,结果如表 5 所示.在干净样本测试中,$NAG_{adv}$-free 相比 PGD-free 略微降低,但在所有类型对抗攻击下的指标上有明显提升.这说明加入 $NAG_{adv}$ 后,模型在面对分类和定位攻击时的防御能力增强.同样地,在 $NAG_{adv}$-LR-free 和 PGD-LR-free 的对比中也表明了 $NAG_{adv}$ 有助于提高模型在对抗攻击场景下的检测性能.对比 PGD-free 和 PGD-LR-free 发现,即使没有 $NAG_{adv}$,LR 也能促进模型鲁棒性的提升.另外,对比 $NAG_{adv}$-LR-free 和 $NAG_{adv}$-free,可以看出 LR 的加入使得分类攻击下的性能提升了 5.18 个百分点(从 42.09%到 47.27%),定位攻击下的性能提升了 4.2 个百分点(从 52.71%到 56.94%).这说明 $NAG_{adv}$ 和 LR 之间存在协同作用,它们能够相互配合显著地提高模型在对抗攻击下的性能.最后综合衡量总训练时长以及对抗防御能力的提升,$NAG_{adv}$ 和 LR 引入增加的及少量训练时间是完全可接受的.

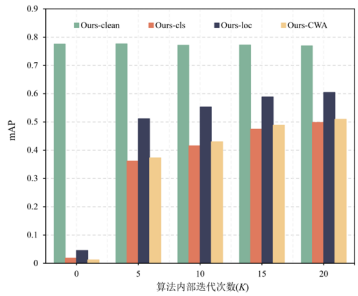
**表 5** 在 PASCAL VOC 2007 测试集上对 $NAG_{adv}$ 和 LR 的消融研究

| 算法 | 干净样本 | $\mathcal{A}_{cls}$ | $\mathcal{A}_{loc}$ | CWA | 训练时间 |
|---|---|---|---|---|---|
| PGD-free | **75.4** | 41.5 | 45.2 | 42.4 | **1462** |
| $NAG_{adv}$-free | 75.38 | **42.09** | **52.71** | **43.87** | 1481 |
| PGD-LR-free | **75.43** | 43.46 | 48.54 | 44.20 | **1468** |
| $NAG_{adv}$-LR-free | 75.39 | **47.27** | **56.94** | **48.81** | 1490 |

其次,考虑了不同模型训练次数($T$)以及算法内部迭代次数($K$)对干净准确率和对抗鲁棒性的影响,在 PASCAL VOC 2007 数据集上对比 PGD 和 $NAG_{adv}$ 在干净样本、定位损失攻击、分类损失攻击和 CWA 攻击下的检测结果如图 3 所示.从图 3(a)中可知,本文 $NAG_{adv}$ 和传统 PGD 训练的模型在对抗攻击干扰下的检测性能均随迭代轮数增加而提高,35000 轮迭代后均达到稳定不再上升,传统 PGD 迭代轮数足够时,能够达到和本文方法相当的性能.区别在于,(1)本文方法训练模型在基于定位对抗攻击下的准确率总是显著优于传统方法;(2)相同迭代次数下,本文方法总是优于传统方法,且能以更少的迭代次数使模型性能达到稳定点,因此与传统方法相比,本文方法起到了加速的作用.从图 3(b)中可知,本文方法的 $K$ 值同样和鲁棒精度呈正相关关系,且基本不会对干净准确率造成影响.



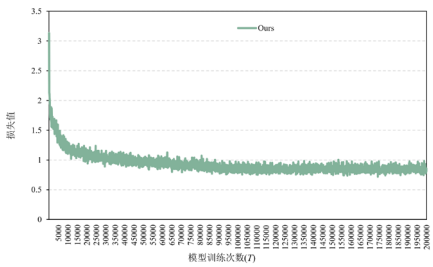

(a)模型训练次数对检测精度影响



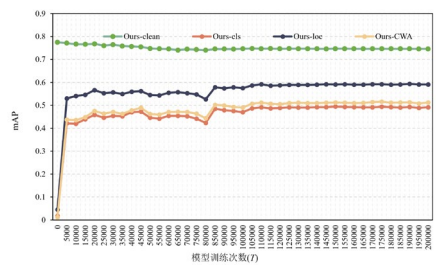

(b) 算法内部迭代次数对检测精度影响

图 3 模型训练次数(*T*)和算法内部迭代次数(*K*)对 PGD 和本文方法所训练模型的检测精度影响

为探讨模型在使用本文算法进行长期训练的表现,我们额外设置了 *T*=200000(5 倍原标准训练时间)进行实验,结果如下图 4 所示.不难看出,训练损失和各项检测精度值都随着迭代增加趋于稳定,没有明显的过拟合或者性能退化等情况发生.同时从图 4(a)也可以看出,模型在经过 20000 轮迭代训练后,损失曲线不再明显下降,说明此时已经收敛.




(a)模型损失变化曲线




(b) 模型检测精度变化曲线

图 4 长期训练下模型的训练损失和检测精度变化趋势图

另外,我们进一步研究了 $NAG_{adv}$ 和 LR 在不同迭代次数($K\in\{10、20\}$)、不同数据集(PASCAL VOC 2007、MS-COCO 2017)上的独立和协同作用,结果如表 6 和表 7 所示.

**表 6** 在 PASCAL VOC 2007 测试集上对不同 *K* 值的 $NAG_{adv}$ 和 LR 的消融研究

| 算法 | 干净样本 | $\mathcal{A}_{cls}$ | $\mathcal{A}_{loc}$ | CWA | 训练时间 |
|---|---|---|---|---|---|
| PGD-10 | **77.21** | 39.44 | 47.03 | 40.27 | **1165** |
| PGD-20 | 76.39 | 43.72 | 50.53 | 43.91 | 2334 |
| PGD-LR-10 | 76.58 | 48.54 | 51.12 | 50.38 | 1167 |
| PGD-LR-20 | 77.19 | 49.43 | 51.92 | 50.04 | 2343 |
| $NAG_{adv}$-10 | 76.95 | 42.19 | 55.52 | 43.97 | 1189 |
| $NAG_{adv}$-20 | 77.21 | 44.70 | 56.61 | 46.21 | 2377 |
| $NAG_{adv}$-LR-10 | 77.10 | 41.55 | 55.27 | 42.99 | 1195 |
| $NAG_{adv}$-LR-20 | 76.90 | **49.84** | **60.47** | **50.93** | 2391 |

**表 7** 在 MS-COCO 2017 验证集上对不同 *K* 值的 $NAG_{adv}$ 和 LR 的消融研究

| 算法 | 干净样本 | $\mathcal{A}_{cls}$ | $\mathcal{A}_{loc}$ | CWA | 训练时间 |
|---|---|---|---|---|---|
| PGD-10 | 40.64 | 1.85 | 8.09 | 2.35 | **2186** |
| PGD-20 | 40.50 | 6.96 | 11.88 | 7.52 | 4378 |
| PGD-LR-10 | 40.42 | 1.79 | 7.98 | 2.05 | 2253 |
| PGD-LR-20 | 40.68 | 11.05 | 13.71 | 11.18 | 4517 |
| $NAG_{adv}$-10 | 40.48 | 13.46 | 20.81 | 15.23 | 2189 |
| $NAG_{adv}$-20 | 40.46 | **16.50** | **22.24** | **17.48** | 4510 |
| $NAG_{adv}$-LR-10 | **40.82** | 16.31 | 20.46 | 17.24 | 2259 |
| $NAG_{adv}$-LR-20 | 40.35 | 15.34 | 20.23 | 16.56 | 4525 |

从表 6 的实验结果来看,$NAG_{adv}$ 和 LR 均能独立提高不同迭代次数基线算法的鲁棒精度.在没有加入 LR 的情况下,$NAG_{adv}$ 能节省约 1 倍的训练时间达到与 PGD 相当的鲁棒性能,同时使用 $NAG_{adv}$ 和 LR 鲁棒精度最高,证明了两者具有一定的协同作用.从表 7 算法在 COCO 数据集的实验结果来看,$NAG_{adv}$ 的鲁棒性提升效果要比 LR 更明显,甚至于两者之间可能存在冲突($NAG_{adv}$-20 优于 $NAG_{adv}$-LR-20),这可能是 COCO 数据集中目标尺度变化大导致的.对于小目标,由于其在图像中占比较小,梯度范数可能相对较小,LR 模块为平衡任务间的损失差异,生成对抗样本时梯度尺度变化可能集中于更易区分的大目标,从而降低了对小目标的关注.相反,$NAG_{adv}$ 可以平衡这些不同尺度目标带来的梯度尺度差异.一方面,动量可以累积小目标的小尺度梯度,使对抗噪声向小目标聚集,从而提高模型对小目标鲁棒特征的学习.同时,对于大目标检测的大尺度梯度,动量可以平滑更新,避免模型过度拟合大目标而忽略小目标.

## 4 总 结

本文提出了一种新的目标检测对抗训练方法,通过引入 NAG 动量加快鲁棒优化内层最大化问题的收敛,达到加速对抗训练的效果.另一方面,在多任务损失函数的基础上,巧妙设计了自适应权重对定位和分类损失进行重加权,实现了目标检测模型鲁棒性的提升,本文的公式推导证明了所提方法与原始 NAG 的收敛速率相匹配,从而建立起优化理论与对抗鲁棒现实场景之间的深层联系,这是一种极其有意义的尝试,并且在实际中取得了比 PGD 更好的效果,对进一步研究对抗鲁棒性的理论收敛具有启发式的意义.此外,希望通过本文的研究,能够促进优化领域中更多先进方法扩展到深度神经网络对抗训练,尤其是应用在目标检测等计算机视觉核心任务中,以提高模型鲁棒性、降低计算开销.

**References**:

[1] Biggio B, Corona I, Maiorca D, Nelson B, Šrndić N, Laskov P, Giacinto G, Roli F. Evasion attacks against machine learning at test time. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD). Berlin, Heidelberg: Springer, 2013. 387–402.

[2] Szegedy C, Zaremba W, Sutskever I, Bruna J, Erhan D, Goodfellow I, Fergus R. Intriguing properties of neural networks. International Conference on Learning Representations (ICLR). 2014.

[3] Goodfellow IJ, Shlens J, Szegedy C. Explaining and harnessing adversarial examples. International Conference on Learning Representations (ICLR). 2015.

[4] Kurakin A, Goodfellow IJ, Bengio S. Adversarial machine learning at scale. International Conference on Learning Representations (ICLR). 2017.

[5] Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. International Conference on Learning Representations (ICLR). 2018.

[6] Tramer F, Kurakin A, Papernot N, Goodfellow I, Boneh D, McDaniel P. Ensemble adversarial training: attacks and defenses. International Conference on Learning Representations (ICLR). 2018.

[7] Zhang H, Yu Y, Jiao J, Xing EP, Ghaoui LE, Jordan MI. Theoretically principled trade-off between robustness and accuracy. International Conference on Machine Learning (ICML). New York, NY: ACM, 2019. 7472–7482.

[8] Wang L, Cao Y, Liu B, Zeng E, Liu K, Xia Y. Ensemble adversarial training defense for time series classification models. Acta Automatica Sinica, 2025, 51(1): 144–160 (in Chinese). DOI: 10.16383/j.aas.c240050.

[9] Samangouei P, Kabkab M, Chellappa R. Defense-GAN: protecting classifiers against adversarial attacks using generative models. International Conference on Learning Representations (ICLR). 2018.

[10] Yoon J, Hwang SJ, Lee J. Adversarial purification with score-based generative models. International Conference on Machine Learning (ICML). New York, NY: ACM, 2021. 12062-12072.

[11] Metzen JH, Genewein T, Fischer V, Bischoff B. On detecting adversarial perturbations. International Conference on Learning Representations (ICLR). 2017.

[12] Pang T, Du C, Dong Y, Zhu J. Towards robust detection of adversarial examples. Annual Conference on Neural Information Processing Systems (NeurIPS). Cambridge, MA: MIT Press, 2018. 4584–4594.

[13] Cohen J, Rosenfeld E, Kolter JZ. Certified adversarial robustness via randomized smoothing. International Conference on Machine Learning (ICML). New York, NY: ACM, 2019. 1310–1320.

[14] Rekavandi AM, Farokhi F, Ohrimenko O, Rubinstein BIP. Certified adversarial robustness via randomized α-smoothing for regression models. Annual Conference on Neural Information Processing Systems (NeurIPS). Cambridge, MA: MIT Press, 2024.

[15] Ma X, Ouyang W, Simonelli A, Ricci E. 3D object detection from images for autonomous driving: a survey. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(5): 3537–3556.

[16] Wang S, Lu S, Cao B. Medical image object detection algorithm for privacy-preserving federated learning. Journal of Computer-Aided Design & Computer Graphics, 2021, 33(10): 1553–1562 (in Chinese). DOI: 10.3724/SP.J.1089.2021.18416.

[17] Lu J, Sibai H, Fabry E. Adversarial examples that fool detectors. arXiv: 1712.02494, 2017.

[18] Xie C, Wang J, Zhang Z, Zhou Y, Xie L, Yuille A. Adversarial examples for semantic segmentation and object detection. IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway, NJ: IEEE, 2017. 1378–1387.

[19] Huang L, Gao C, Zhou Y, Xie C, Yuille AL, Zou C, Liu N. Universal physical camouflage attacks on object detectors. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2020. 717–726.

[20] Liang S, Wu B, Fan Y, Wei X, Cao X. Parallel rectangle flip attack: a query-based black-box attack against object detection. IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway, NJ: IEEE, 2021. 7677–7687.

[21] Cai Z, Xie X, Li S, Yin M, Song C, Krishnamurthy SV, Roy-Chowdhury AK, Asif MS. Context-aware transfer attacks for object detection. AAAI Conference on Artificial Intelligence (AAAI), Menlo Park, CA: AAAI, 2022. 149–157.

[22] Huang H, Chen Z, Chen H, Wang Y, Zhang K. T-sea: transfer-based self-ensemble attack on object detection. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2023. 20514–20523.

[23] Wei X, Liang S, Chen N, Cao X. Transferable adversarial attacks for image and video object detection. International Joint Conference on Artificial Intelligence (IJCAI). San Francisco, CA: Morgan Kaufmann , 2019. 954–960.

[24] Lu YX, Liu ZY, Luo YG, Deng SY, Jiang T, Ma JY, Dong YP. Black-box Transferable Attack Method for Object Detection Based on GAN. Ruan Jian Xue Bao/Journal of Software, 2024, 35(7): 3531–3550 (in Chinese). http://www.jos.org.cn/1000-9825/6937.htm.

[25] Wang X, Chen J, He K, Zhang Z, Du R, Li Q, She J. Survey on adversarial attacks and defenses for object detection. Journal on communications, 2023, 44(11): 260–277 (in Chinese). DOI: 10.11959/j.issn.1000-436x.2023223.

[26] Zhang H, Wang J. Towards adversarially robust object detection. IEEE/CVF International Conference on Computer Vision (ICCV). Piscataway, NJ: IEEE, 2019. 421–430.

[27] Chen P-C, Kung B-H, Chen J-C. Class-aware robust adversarial training for object detection. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Piscataway, NJ: IEEE, 2021. 10415–10424.

[28] Dong Z, Wei P, Lin L. Adversarially-aware robust object detector. European Conference on Computer Vision (ECCV). Berlin: Springer, 2022. 297–313.

[29] Nesterov Y. A method of solving a convex programming problem with convergence rate $O(1/k^2)$. Soviet Mathematics Doklady, 1983, 27(2): 372–376.

[30] Long S, Tao W, Zhang ZD, Tao Q. Adaptive NAG Methods Based on AdaGrad and Its Optimal Individual Convergence. Ruan Jian Xue Bao/Journal of Software, 2022, 33(4): 1231–1243 (in Chinese).

[31] Xie X, Zhou P, Li H, Lin Z, Yan S. Adan: adaptive nesterov momentum algorithm for faster optimizing deep models. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(12): 9508–9520.

[32] Lin J, Song C, He K, Wang L, Hopcroft JE. Nesterov accelerated gradient and scale invariance for adversarial attacks. International Conference on Learning Representations (ICLR). 2020.

[33] Zou J, Duan Y, Ren C, Qiu J, Zhou X, Pan Z. Perturbation Initialization, Adam-Nesterov and Quasi-Hyperbolic Momentum for Adversarial Examples. Acta Electronica Sinica, 2022, 50(1): 207–216 (in Chinese).

[34] Bao L, Tao W, Tao Q. Boosting Adversarial Transferability Through Adaptive-Learning-Rate with Data Augmentation Mechanism. Acta Electronica Sinica, 2024, 52(1): 157–169 (in Chinese).

[35] Shafahi A, Najibi M, Ghiasi MA, Xu Z, Dickerson J, Studer C, Davis LS, Taylor G, Goldstein T. Adversarial training for free! Annual Conference on Neural Information Processing Systems (NeurIPS). Cambridge, MA: MIT Press, 2019. 3353–3364.

[36] Wong E, Rice L. FAST is better than free: revisiting adversarial training. International Conference on Learning Representations (ICLR). 2020.

[37] Jia X, Zhang Y, Wei X, Wu B, Ma K, Wang J, Cao X. Prior-guided adversarial initialization for fast adversarial training. Avidan S, Brostow G, Cissé M, Farinella G M, Hassner T. European Conference on Computer Vision (ECCV). Berlin: Springer, 2022. 567–584.

[38] Jia X, Zhang Y, Wei X, Wu B, Ma K, Wang J, Cao X. Improving fast adversarial training with prior-guided knowledge. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(9): 6367–6383.

[39] Nesterov Y. Introductory Lectures on Convex Optimization - A Basic Course. Applied Optimization 87, Springer 2004, ISBN 978-1-4613-4691-3, pp. 1–236.

[40] Everingham M, Eslami SMA, Van Gool L, Williams CKI, Winn J, Zisserman A. The pascal visual object classes challenge: a retrospective. International Journal of Computer Vision, 2015, 111(1): 98–136.

[41] Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft coco: common objects in context. Fleet D, Pajdla T, Schiele B, Tuytelaars T. European Conference on Computer Vision (ECCV). Berlin: Springer, 2014. 740–755.

[42] Ghadimi S, Lan G. Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Mathematical Programming, 2016, 156(1–2): 59–99.

**附中文参考文献**:

[8]王璐瑶, 曹渊, 刘博涵, 曾恩, 刘坤, 夏元清. 时间序列分类模型的集成对抗训练防御方法. 自动化学报, 2025, 51(1): 144–160. DOI: 10.16383/j.aas.c240050

[16]王生生, 路淑贞, 曹斌. 面向隐私保护联邦学习的医学影像目标检测算法. 计算机辅助设计与图形学学报, 2021, 33(10): 1553–1562. DOI: 10.3724/SP.J.1089.2021.18416

[24]陆宇轩, 刘泽禹, 罗咏刚, 邓森友, 江天, 马金燕, 董胤蓬. 基于生成对抗网络的目标检测黑盒迁移攻击算法. 软件学报, 2024, 35(7): 3531–3550. http://www.jos.org.cn/1000-9825/6937.htm

[25]汪欣欣, 陈晶, 何琨, 张子君, 杜瑞颖, 李瞧, 佘计思. 面向目标检测的对抗攻击与防御综述. 通信学报, 2023,44(11):260–277. DOI: 10.11959/j.issn.1000-436x.2023223.

[30]陇盛, 陶蔚, 张泽东, 陶卿. 基于 AdaGrad 的自适应 NAG 方法及其最优个体收敛性. 软件学报, 2022, 33(4):1231–1243. http://www.jos.org.cn/1000-9825/6464.htm

[33]邹军华, 段晔鑫, 任传伦, 邱俊洋, 周星宇, 潘志松. 基于噪声初始化、Adam-Nesterov 方法和准双曲动量方法的对抗样本生成方法[J]. 电子学报, 2022, 50(1): 207–216. https://doi.org/10.12263/DZXB.20200839.

[34]鲍蕾, 陶蔚, 陶卿. 结合自适应步长策略和数据增强机制提升对抗攻击迁移性. 电子学报, 2024, 52(1): 157-169.

## 附录 A 定理 1 证明

若 $\mathcal{S}_{\boldsymbol{x}}=\mathbb{R}^d$ ,那么有

$$\bar{\boldsymbol{x}}_k=\arg\min_{\boldsymbol{z}\in\mathcal{S}_{\boldsymbol{x}}}\left\{F\left(\boldsymbol{x}_k\right)+\left\langle F'\left(\boldsymbol{x}_k\right),\boldsymbol{z}-\boldsymbol{x}_k\right\rangle+\frac{1}{2\eta_k}\left\|\boldsymbol{z}-\boldsymbol{x}_k\right\|^2\right\}\Leftrightarrow\bar{\boldsymbol{x}}_k=\boldsymbol{x}_k-\eta_kF'\left(\boldsymbol{x}_k\right)$$

因此 $\mathrm{NAG}_{\mathrm{adv}}$ 迭代公式可表示为:

$$\begin{cases}a_{k+1}=\left(1+\sqrt{4a_k^2+1}\right)\Big/2\\ \boldsymbol{x}_{k+1}=\bar{\boldsymbol{x}}_k+\left(a_k-1\right)\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}\right)\Big/a_{k+1} & ,\forall k\geq 0\\ \bar{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_{k+1}-\eta_{k+1}F'\left(\boldsymbol{x}_{k+1}\right)\end{cases}\tag{1}$$

因为 $\sigma_{k+1}\equiv 1/a_{k+1}$ ,则 $\sigma_k\equiv 1/a_k$ , $\sigma_k^2\equiv 1/a_k^2$ , $\sigma_{k+1}\left(\sigma_k^{-1}-1\right)\equiv\left(a_k-1\right)/a_{k+1}$ .代入(1)式第二行得

$$\begin{aligned}\boldsymbol{x}_{k+1}&=\bar{\boldsymbol{x}}_k+\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}\right)\sigma_{k+1}\left(\sigma_k^{-1}-1\right)\\&=\bar{\boldsymbol{x}}_k-\sigma_{k+1}\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_k\sigma_k^{-1}-\bar{\boldsymbol{x}}_{k-1}+\bar{\boldsymbol{x}}_{k-1}\sigma_k^{-1}\right)\\&=\bar{\boldsymbol{x}}_k-\sigma_{k+1}\left(\bar{\boldsymbol{x}}_k-\sigma_k^{-1}\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}\left(1-\sigma_k\right)\right)\right)\end{aligned}\tag{2}$$

令 $\boldsymbol{z}_k\equiv\sigma_k^{-1}\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}\left(1-\sigma_k\right)\right)$ 有

$$\bar{\boldsymbol{x}}_k=\left(1-\sigma_k\right)\bar{\boldsymbol{x}}_{k-1}+\sigma_k\boldsymbol{z}_k,\quad \bar{\boldsymbol{x}}_{k+1}=\left(1-\sigma_{k+1}\right)\bar{\boldsymbol{x}}_k+\sigma_{k+1}\boldsymbol{z}_{k+1}\tag{3}$$

将 $\boldsymbol{z}_k\equiv\sigma_k^{-1}\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}\left(1-\sigma_k\right)\right)$ 代入(2)式得

$$\boldsymbol{x}_{k+1}=\left(1-\sigma_{k+1}\right)\bar{\boldsymbol{x}}_k+\sigma_{k+1}\boldsymbol{z}_k\tag{4}$$

联立(3)(4)式可得

$$\boldsymbol{x}_{k+1}=\bar{\boldsymbol{x}}_{k+1}-\sigma_{k+1}\boldsymbol{z}_{k+1}+\sigma_{k+1}\boldsymbol{z}_k=\bar{\boldsymbol{x}}_{k+1}-\sigma_{k+1}\left(\boldsymbol{z}_{k+1}-\boldsymbol{z}_k\right)\tag{5}$$

上式移项,方程两边同时除以 $\sigma_{k+1}$ 得

$$\boldsymbol{z}_{k+1}-\boldsymbol{z}_k=\sigma_{k+1}^{-1}\bar{\boldsymbol{x}}_{k+1}-\sigma_{k+1}^{-1}\boldsymbol{x}_{k+1}\tag{6}$$

将(1)式第三行代入(6)式得

$$\boldsymbol{z}_{k+1}=\boldsymbol{z}_k-\sigma_{k+1}^{-1}\eta_{k+1}F'\left(\boldsymbol{x}_{k+1}\right)\tag{7}$$

结合(3)(4)(7)式可得

$$\begin{cases}\boldsymbol{x}_{k+1}=\left(1-\sigma_{k+1}\right)\bar{\boldsymbol{x}}_k+\sigma_{k+1}\boldsymbol{z}_k\\ \boldsymbol{z}_{k+1}=\boldsymbol{z}_k-\sigma_{k+1}^{-1}\eta_{k+1}F'\left(\boldsymbol{x}_{k+1}\right)\\ \bar{\boldsymbol{x}}_{k+1}=\left(1-\sigma_{k+1}\right)\bar{\boldsymbol{x}}_k+\sigma_{k+1}\boldsymbol{z}_{k+1}\end{cases}\quad,\forall k\geq 0\tag{8}$$

定理 1 得证.

## 附录 B 定理 2 证明

同定理 1 证明,$\mathrm{NAG}_{\mathrm{adv}}$ 迭代公式等价于:

$$\begin{cases}a_{k+1}=\left(1+\sqrt{4a_k^2+1}\right)/2\\ \boldsymbol{x}_{k+1}=\bar{\boldsymbol{x}}_k+\left(a_k-1\right)\left(\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}\right)/a_{k+1}\\ \bar{\boldsymbol{x}}_{k+1}=\boldsymbol{x}_{k+1}-\eta_{k+1}F'\left(\boldsymbol{x}_{k+1}\right)\end{cases}\quad,\forall k\geq 0\tag{1}$$

令 $\boldsymbol{v}_k\equiv\bar{\boldsymbol{x}}_k-\bar{\boldsymbol{x}}_{k-1}$ $\lambda_k\equiv\left(a_k-1\right)/a_{k+1}$ .(1)式第二行变为

$$\boldsymbol{x}_{k+1}=\overline{\boldsymbol{x}}_k+\lambda_k\boldsymbol{v}_k \tag{2}$$

将(2)式代入(1)式第三行移项得

$$\overline{\boldsymbol{x}}_{k+1}-\overline{\boldsymbol{x}}_k=\lambda_k\boldsymbol{v}_k-\eta_{k+1}F'\left(\overline{\boldsymbol{x}}_k+\lambda_k\boldsymbol{v}_k\right) \tag{3}$$

将 $\boldsymbol{v}_{k+1}\equiv\overline{\boldsymbol{x}}_{k+1}-\overline{\boldsymbol{x}}_k$ 与(3)式联立得

$$\begin{cases}\boldsymbol{v}_{k+1}=\lambda_k\boldsymbol{v}_k-\eta_{k+1}F'\left(\overline{\boldsymbol{x}}_k+\lambda_k\boldsymbol{v}_k\right)\\ \overline{\boldsymbol{x}}_{k+1}=\overline{\boldsymbol{x}}_k+\boldsymbol{v}_{k+1}\end{cases} \tag{4}$$

令 $\boldsymbol{g}_k\equiv\eta_k^{-1}\boldsymbol{v}_k$ ,(4)式可改写为

$$\begin{cases}\boldsymbol{g}_{k+1}=\lambda_k\eta_k\eta_{k+1}^{-1}\boldsymbol{g}_k-F'\left(\overline{\boldsymbol{x}}_k+\lambda_k\eta_k\boldsymbol{g}_k\right)\\ \overline{\boldsymbol{x}}_{k+1}=\overline{\boldsymbol{x}}_k+\eta_{k+1}\boldsymbol{g}_{k+1}\end{cases} \tag{5}$$

令 $\overline{\boldsymbol{x}}_k^{\text{nes}}\equiv\overline{\boldsymbol{x}}_k+\lambda_k\eta_k\boldsymbol{g}_k$ , $\mu_k\equiv\lambda_k\eta_k\eta_{k+1}^{-1}$ .(5)式可改写为

$$\begin{cases}\overline{\boldsymbol{x}}_k^{\text{nes}}\equiv\overline{\boldsymbol{x}}_k+\eta_{k+1}\mu_k\boldsymbol{g}_k\\ \boldsymbol{g}_{k+1}=\mu_k\boldsymbol{g}_k-F'\left(\overline{\boldsymbol{x}}_k^{\text{nes}}\right)\\ \overline{\boldsymbol{x}}_{k+1}=\overline{\boldsymbol{x}}_k+\eta_{k+1}\boldsymbol{g}_{k+1}\end{cases}\quad,\forall k\geq 0 \tag{6}$$

综上, $\mu_k\equiv\eta_k\eta_{k+1}^{-1}(a_k-1)/a_{k+1}$ ,命题 2 得证.