On the Convergence of an Adaptive Momentum Method for Adversarial Attacks

Sheng Long[1*], **Wei Tao**[1,2*], **Shuohao Li**[1], **Jun Lei**[1], **Jun Zhang**[1†]

[1]Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China
[2]Strategic Assessments and Consultation Institute, Academy of Military Science, Beijing 100091, China
{longsheng, lishuohao, leijun1987, zhangjun1975}@nudt.edu.cn, wtao_plaust@163.com

Abstract

Adversarial examples are commonly created by solving a constrained optimization problem, typically using sign-based methods like Fast Gradient Sign Method (FGSM). These attacks can benefit from momentum with a constant parameter, such as Momentum Iterative FGSM (MI-FGSM), to enhance black-box transferability. However, the monotonic time-varying momentum parameter is required to guarantee convergence in theory, creating a theory-practice gap. Additionally, recent work shows that sign-based methods fail to converge to the optimum in several convex settings, exacerbating the issue. To address these concerns, we propose a novel method which incorporates both an innovative adaptive momentum parameter without monotonicity assumptions and an adaptive step-size scheme that replaces the sign operation. Furthermore, we derive a regret upper bound for general convex functions. Experiments on multiple models demonstrate the efficacy of our method in generating adversarial examples with human-imperceptible noise while achieving high attack success rates, indicating its superiority over previous adversarial example generation methods.

Introduction

Deep neural networks are known to be vulnerable to adversarial examples, which are imperceptible to the human eye but mislead the classifier (Szegedy et al. 2014)(Goodfellow, Shlens, and Szegedy 2015). Adversarial examples play a key role in improving model robustness through adversarial training (Madry et al. 2018)(Shafahi et al. 2019)(Pang et al. 2020)(Cai et al. 2021), thus offering a defense mechanism against unknown adversarial attacks. Generally speaking, crafting high-quality adversarial examples has garnered considerable attention. The generation process is typically formulated as a constrained optimization problem and solved using sign-based methods such as FGSM (Goodfellow, Shlens, and Szegedy 2015), Basic Iterative Method (BIM)(Kurakin, Goodfellow, and Bengio 2017), Project Gradient Descent (PGD) (Madry et al. 2018), etc.

To further boost the attack success rate on black-box models, an effective strategy involves training adversarial examples with transferability on a white-box surrogate model (Papernot, McDaniel, and Goodfellow 2016; Papernot et al. 2017)(Xie et al. 2019). Notably, representative algorithms derived from FGSM, namely MI-FGSM (Dong et al. 2018) and Nesterov Iterative FGSM (NI-FGSM) (Lin et al. 2020), correspond to Heavy-Ball (HB) (Polyak 1964) and Nesterov Accelerated Gradient (NAG) (Nesterov 1983), respectively. In particular, MI-FGSM is highly similar to HB, and we will delve into the relationship between them in the next section. HB (also named SGD with momentum) with a constant momentum parameter has been widely adopted in practice to improve the generalization performance of deep learning models (Sutskever et al. 2013)(Goyal et al. 2017). HB has also shown evidence of being helpful in dampening oscillations (Ruder 2016), and escaping local minimum or saddle points (Ochs et al. 2014)(Sun et al. 2019). Inheriting these benefits from HB, MI-FGSM achieves stability in the update direction, generating adversarial examples with stronger transferability.

However, from an optimization perspective, the monotonic time-varying momentum parameter is needed for the convergence analysis. There are two proposals for momentum parameter to guarantee convergence, but completely opposite. Specifically, one assumes a strictly monotonically decreasing schedule ($\beta_{1,t-1} > \beta_{1,t}, \beta_{1,t} \to 0$) (Kingma and Ba 2015)(Reddi, Kale, and Kumar 2018)(Wang et al. 2020)(Zhuang et al. 2020) while the other demands an increasing schedule ($\beta_{1,t-1} < \beta_{1,t}, \beta_{1,t} \to 1$) (Ghadimi, Feyzmahdavian, and Johansson 2015)(Yang, Lin, and Li 2016)(Tao et al. 2021)(Li, Liu, and Orabona 2022), which not only creates a theory-practice gap but also causes confusion when selecting hyper-parameters in practice. On the other hand, (Ghadimi, Feyzmahdavian, and Johansson 2015) utilizes a constant momentum parameter, which guarantees the convergence of HB, but relies on strong assumptions of strong convexity and smoothness in the objective function. (Alacaoglu et al. 2020) proposes a novel theoretical analysis framework for Adam-type methods and obtains data-dependent regret bounds with a constant momentum parameter β_1, yet, their bound still falls short of the optimal upper bound with a logarithmic factor gap.

Faced with these challenges, it is natural to question: (1) Why not investigate the convergence of MI-FGSM using the theoretical results of HB? (2) Why not directly use HB, which guarantees convergence, to generate adversarial ex-

[*]These authors contributed equally.

[†]Corresponding Author.

amples? We provide the following two analyses:

The regret bound of HB is insufficient for analyzing the convergence of MI-FGSM under constrained cases. In other words, MI-FGSM cannot be equivalently converted to HB due to the involved coupling among projection (clipping) operation, sign function, momentum parameter, stepsize parameter and gradient normalization. We will thoroughly explore the differences between MI-FGSM and HB in the next section.

Directly employing vanilla HB without the sign function may not yield high-quality adversarial examples. Empirical evidence indicates that the use of the sign function plays a significant role in algorithm performance improvement (Kunstner et al. 2023)(Chen et al. 2023). Meanwhile, (Liu et al. 2019) also shows that the generation of adversarial examples can be interpreted by signSGD, which compresses gradient scales based on the sign function, effectively mitigating the negative effects of noise.

In contrast, (Karimireddy et al. 2019) shows that sign-based methods fail to converge to the optimum in several convex settings, limiting optimal convergence analysis to non-convex (Bernstein et al. 2018) and smooth (Crawshaw et al. 2022) environments. (Gao et al. 2021) find that sign-based methods only extract the sign of gradient units but ignore their value difference, which inevitably leads to a deviation. Moreover, there is also evidence demonstrating that sign-based methods seem to have an adverse effect on the generalization performance of the obtained solutions (Balles and Hennig 2018).

Fortunately, sign-based methods are inextricably linked to Adam-type methods (Bernstein et al. 2018)(Zhuang et al. 2020)(Tao et al. 2023). Therefore, the adaptive step-size can serve as an alternative way to evade the aforementioned conflicting theoretical views regarding the sign function. Based on this insight, our contributions are as follows:

- We propose AdaMSI-FGM, which introduces an adaptive momentum parameter under weaker assumptions to bridge the theory-practice gap mentioned above, and incorporates an adaptive step-size scheme to evade the potential negative effects of sign function.
- In the non-smooth convex setting, we derive a data-dependent regret bound $O(\sqrt{T})$, with a slight improvement in the suboptimal regret bound $O(\sqrt{\log(T)T})$ obtained by the traditional Adam-type algorithms.
- Through extensive evaluations on diverse models, we demonstrate the effectiveness of our approach in generating adversarial examples with higher attack success rates and human-imperceptible noises.

Related Work

We aim to generate a non-targeted adversarial example, denoted as $\boldsymbol{x}$, from a clean example $\boldsymbol{x}_0$ with the true label y. The generated adversarial example should satisfy the L_∞ norm bound constraint. This process can be formulated as a constrained optimization problem presented below:

$$\min f(\boldsymbol{x}), \quad s.t.\|\boldsymbol{x}-\boldsymbol{x}_0\|_\infty \le \epsilon, \tag{1}$$

where $f(\boldsymbol{x}) = -J(\boldsymbol{x}, y)$ for simplicity, and we can learn adversarial example $\boldsymbol{x}$ directly using the gradient descent direction instead of the gradient ascent direction. J represents the loss function, typically the cross-entropy loss. ϵ denotes the size of adversarial perturbation. $\boldsymbol{x}^*$ is one of the optimal solutions.

Preliminaries. We use lower case letters to denote scalars, lower case bold face letters to denote vectors, upper case bold face letters to denote matrices. For any vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^d$, all standard operations such as $\boldsymbol{ab}$, $\boldsymbol{a}^2$, $\boldsymbol{a}^{\frac{1}{2}}$, $\frac{1}{\boldsymbol{a}}$ are assumed to be element-wise. We use $\operatorname{diag}(\boldsymbol{a})$ to denote a $d \times d$ matrix which has $\boldsymbol{a}$ in its diagonal, and the rest of its element are all 0. We denote the L_p norm ($p \ge 1$) of $\boldsymbol{a}$ by $\|\boldsymbol{a}\|_p = (\sum_{i=1}^d |a_i|^p)^{\frac{1}{p}}$, the L_∞ norm of $\boldsymbol{a}$ by $\|\boldsymbol{a}\|_\infty = \max_{i=1}^d |a_i|$ For a sequence of vectors $\{\boldsymbol{a}_t\}_{t=1}^T$, we denote the i^{th} element of $\boldsymbol{a}_t$ by $a_{t,i}$. For a sequence of diagonal matrices $\{\boldsymbol{A}_t\}_{t=1}^T$, we use $A_{t,i}$ to denote the i^{th} element in the diagonal of $\boldsymbol{A}_t$. We use $\nabla f(\boldsymbol{x}_t)$ to denote the gradient of $f(\cdot)$ at $\boldsymbol{x}_t$, further writing as $\boldsymbol{g}_t$ for simplicity. We also use $\boldsymbol{g}_{1:t,i} = [g_{1,i}, ..., g_{t,i}]$ to denote the vector obtained by concatenating the i^{th} element of the gradient sequence.

Sign-Based Methods for Generating Adversarial Examples

A large amount of sign-based attack methods have been proposed to solve Problem 1 in past years. FGSM (Goodfellow, Shlens, and Szegedy 2015) is one of the earliest methods, which generates an adversarial example $\boldsymbol{x}$ using a one-step update:

$$\boldsymbol{x} = \boldsymbol{x}_0 - \epsilon \cdot \operatorname{sign}(\nabla f(\boldsymbol{x}_0)). \tag{FGSM}$$

However, even with prior knowledge of the model's structure, the adversarial examples generated by FGSM often fail to achieve high attack success rates. To tackle this issue, (Kurakin, Goodfellow, and Bengio 2017) propose BIM, which is an iterative extension of FGSM:

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_T \cdot \operatorname{sign}(\nabla f(\boldsymbol{x}_t)), \tag{BIM}$$

where $\alpha_T = \frac{\epsilon}{T}$. To further improve the attack success rates on white-box models, (Madry et al. 2018) propose PGD by incorporating random noise initialization into BIM. However, improving the success rate for white-box attacks comes at the cost of decreased success rates for black-box attacks. To address this trade-off, (Dong et al. 2018) introduce MI-FGSM, which utilizes momentum to enhance transferability of adversarial examples without sacrificing performance on white-box models:

$$\begin{aligned} \boldsymbol{m}_t &= \mu \boldsymbol{m}_{t-1} + \frac{\nabla f(\boldsymbol{x}_t)}{\|\nabla f(\boldsymbol{x}_t)\|_1}, \\ \boldsymbol{x}_{t+1} &= \boldsymbol{x}_t - \alpha_T \cdot \operatorname{sign}(\boldsymbol{m}_t), \end{aligned} \tag{MI-FGSM}$$

where $\boldsymbol{m}_{-1} = \boldsymbol{0}$ and the momentum parameter μ is suggested as a constant value. Under the intuition of using better optimization methods to generate adversarial examples, (Lin et al. 2020) modify the gradient calculation position of MI-FGSM, and propose NI-FGSM to enhance transferability. Nevertheless, there is still no convergence analysis for both MI-FGSM and NI-FGSM.

Momentum Methods with Convergence Guarantee

MI-FGSM is inspired by the concept of momentum, which can be traced back to HB method (Polyak 1964). The iterates of HB are given by

$$\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_t \nabla f(\boldsymbol{x}_t) + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1}), \qquad \textbf{(HB)}$$

where $\boldsymbol{x}_{-1} = \boldsymbol{x}_0$, $\alpha_t = \frac{\alpha}{\sqrt{t}}$, $\alpha > 0$. Compared to the update rules of SGD: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t - \alpha_t \nabla f(\boldsymbol{x}_t)$, HB adds a momentum term: $\beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})$, which allows the algorithm to maintain inertia when the gradient is small and overcome the obstacles such as saddle points and local optima. The constant parameter $\beta_t \equiv (\sqrt{L} - \sqrt{\mu}/\sqrt{L} + \sqrt{\mu})^2$ is recommended to achieve a linear convergence rate, while the knowledge of the Lipschitz constant L and strongly convex coefficient μ are generally inaccessible. In order to obtain the optimal convergence rate under general convex environments, (Ghadimi, Feyzmahdavian, and Johansson 2015) introduced a monotonically increasing momentum parameter schedule $\beta_t = t/(t+2)$. Furthermore, (Tao, Wu, and Tao 2022) generalized this setting to non-smooth convex case.

In deep neural network literature, however, HB method is more often rewritten as SGD with Momentum (SGDM) (Sutskever et al. 2013), which is widely used in deep learning libraries like PyTorch and TensorFlow. The update procedure is formalized as follows:

$$\begin{aligned} \boldsymbol{m}_t &= \hat{\beta}_t \boldsymbol{m}_{t-1} + \nabla f(\boldsymbol{x}_t), \\ \boldsymbol{x}_{t+1} &= \boldsymbol{x}_t - \alpha_t \boldsymbol{m}_t, \end{aligned} \qquad \textbf{(SGDM)}$$

where $\boldsymbol{m}_{-1} = \boldsymbol{0}$. It is worth noting that HB can be one-to-one mapped to SGDM by setting $\hat{\beta}_t = \frac{\alpha_{t-1}}{\alpha_t}\beta_t$ under unconstrained cases, which indicates that a time-varying momentum parameter is indispensable for SGDM to inherit the convergence results of HB. Meanwhile, another momentum technique named First Moment Estimate (FME) is widely used in Adam-type algorithms. FME is obtain by removing the second moment estimate in Adam (Kingma and Ba 2015):

$$\begin{aligned} \boldsymbol{m}_t &= \beta_{1,t} \boldsymbol{m}_{t-1} + (1-\beta_{1,t})\nabla f(\boldsymbol{x}_t), \\ \boldsymbol{x}_{t+1} &= \boldsymbol{x}_t - \alpha_t \boldsymbol{m}_t, \end{aligned} \qquad \textbf{(FME)}$$

where $\boldsymbol{m}_{-1} = \boldsymbol{0}$. Unlike HB, FME does not have an equivalent conversion relation with SGDM, making the convergence analysis of HB not applicable to FME. Adam variants require a decreasing $\beta_{1,t} \to 0$ schedule to derive $O(\sqrt{T})$ regret bound in theory, but a constant $\beta_{1,t} \equiv 0.9$ is commonly used in practice. (Alacaoglu et al. 2020) bridge this gap by deriving suboptimal $O(\sqrt{\log(T)T})$ regret bound with a constant β_1. However, (Li, Liu, and Orabona 2022) demonstrate that the last iterate of FME can only obtain a suboptimal convergence rate for any constant momentum parameter β, which alone kills the possibility of any advantage of FME with constant momentum, unless stronger assumptions are used.

Remark. The generation of adversarial examples using MI-FGSM can be seen as analogous to training neural network models with HB (or SGDM). There is a clear connection between MI-FGSM and SGDM in their update rules, with the main difference being the use of the sign function and gradient normalization. While MI-FGSM can be approximately equivalent to HB under unconstrained cases, the convergence analysis of MI-FGSM is significantly hindered in the adversarial attack Problem 1 with box constraints, particularly due to the coupling between the projection (clipping) operation and the sign function.

Methodology

To address the theory-practice gap caused by the momentum parameter and overcome the non-convergence issue of sign-based methods in convex cases, we propose AdaMSI-FGM. This novel approach combines adaptive HB momentum parameter and adaptive step-size to generate adversarial examples.

Adaptation of Momentum Parameter

The two existing proposals of momentum parameters assume monotonic time-varying schedule β_t to guarantee convergence. However, a constant momentum parameter β is generally used in practice. To bridge this gap, we introduce a novel adaptive parameter $\beta_{1,t}$ for HB momentum without the monotonicity assumption, which is well-designed as follows

$$\beta_{1,t} = s_{t-1}/(s_t + 1), \tag{2}$$

$$s_t = \lambda^{\frac{t}{2}} \|\boldsymbol{g}_t\|_1, \tag{3}$$

where the structure of Equation 2 is inspired by the theoretical analysis of traditional HB momentum methods (Ghadimi, Feyzmahdavian, and Johansson 2015), and Equation 3 relies on the hyper-parameter $\lambda \in (0,1)$ and the L_1 norm of gradient information. By incorporating Equation 3 into Equation 2, we obtain

$$\beta_{1,t} = \frac{\lambda^{\frac{t-1}{2}} \|\boldsymbol{g}_{t-1}\|_1}{\lambda^{\frac{t}{2}} \|\boldsymbol{g}_t\|_1 + 1}. \tag{4}$$

From Equation 4, it can be observed that the proposed momentum parameter $\beta_{1,t}$ automatically adapts to the real-time variation in the coupling of the current gradient, the latest gradient, and the time-step. This provides a more general and flexible condition compared to the monotonicity assumption and resolves the challenge of momentum parameter selection in practice.

Adaptation of Step-Size

MI-FGSM has shown remarkable effectiveness in adversarial attacks. However, from a theoretical point of view, there are still gaps in our knowledge, particularly regarding the inability of sign-based methods to converge to the optimum in convex environments (Karimireddy et al. 2019). To tackle this issue, we found a breakthrough from the connection between sign function and adaptive step-size.

Adaptive step-size has attracted widespread attention as its benefits for sparse optimization (Duchi, Hazan, and Singer 2010) and its ability to yield tighter data-dependent regret bounds (Kingma and Ba 2015)(Reddi, Kale, and Kumar 2018). Correspondingly to previously mentioned FME, adaptive step-size is also referred to as Second Moment Estimate (SME), which can be formalized as follows

$$\boldsymbol{v}_t = \beta_{2,t}\boldsymbol{v}_{t-1} + (1-\beta_{2,t})\boldsymbol{g}_t^2, \qquad \textbf{(SME)}$$

where $\beta_{2,t}$ is generally chosen from $\{0.99, 0.999\}$ in practice. From a theoretical perspective, setting $1-\frac{1}{t} \le \beta_{2,t} \le 1-\frac{\gamma}{t}$, $0 < \gamma \le 1$, effectively solves the divergence problem of Adam as proposed by (Reddi, Kale, and Kumar 2018). In particular, when setting $\gamma = 1$, that is $\beta_{2,t} = 1-\frac{1}{t}$, we have

$$\boldsymbol{v}_t = \frac{1}{t}\sum_{j=1}^{t}\boldsymbol{g}_j^2. \tag{5}$$

According to the above settings, Adam with both FME and SME degenerates into AdaGrad (Duchi, Hazan, and Singer 2010) with FME. One potential advantage of adaptive step-size methods is the "sparse noise reduction" effect highlighted by (Bernstein et al. 2018), more importantly, they introduce an intriguing connection between Adam and signSGD:

$$\text{Adam} \sim \text{FME/SME} = \frac{\boldsymbol{m}_t}{\sqrt{\boldsymbol{v}_t}}, \tag{6}$$

$$\text{signSGD} \sim \text{sign}(\boldsymbol{g}_t) = \frac{\boldsymbol{g}_t}{\sqrt{\boldsymbol{g}_t^2}}. \tag{7}$$

Based on this fact, setting the time-varying parameters in FME and SME to zero ($\beta_{1,t} \to 0$ and $\beta_{2,t} \to 0$) results in Adam being converted to signSGD. (Zhuang et al. 2020) also report that update direction in Adam is close to "sign descent" in low-variance case. It is worth noting that even though the sign function is a key motivation behind Adam-type methods, Adam still outperforms sign-based methods in full batch scenarios (Kunstner et al. 2023). What's worse, (Karimireddy et al. 2019) demonstrated that sign-based methods fail to converge to the optimum in several convex settings. As a result, we suggest using SME, the most representative adaptive step-size scheme, as a replacement for the key role of a sign function in adversarial attacks.

By integrating the adaptive momentum parameter and adaptive step-size together, the proposed AdaMSI-FGM is summarized in Algorithm 1.

Convergence Analysis

To solve the constrained optimization Problem 1 and further complete the convergence analysis, our AdaMSI-FGM can be simplified as

$$\boldsymbol{x}_{t+1} = P_{\boldsymbol{\chi}}\left\{\boldsymbol{x}_t - \alpha_t\hat{\boldsymbol{V}}_t^{-1}\boldsymbol{g}_t + \beta_t(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right\}, \tag{8}$$

where $P_{\boldsymbol{\chi}}$ denotes projection operator onto the convex set $\boldsymbol{\chi} : \{\boldsymbol{x} | \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty \le \epsilon\}$ and plays the same role as clipping operation. Under such a situation, it is not difficult to observe that

$$\|\boldsymbol{x}_t - \boldsymbol{x}^*\|_\infty \le \epsilon, \tag{9}$$

where $\{\boldsymbol{x}_t\}$ generated by Equation 8. Moreover, we also need to provide an assumption that has been prevalent in previous convergence analysis, as shown below.

Algorithm 1: Adaptive Momentum and Step-size Iterate Fast Gradient Method (AdaMSI-FGM)

```
Input: $\boldsymbol{x}_0$; $\{\alpha_t\}_{t=1}^T$; $\{\beta_{2,t}\}_{t=1}^T$; $\{\xi_t\}_{t=1}^T$; $\epsilon$; $\lambda$.
Initialize: $\boldsymbol{x}_0 = \boldsymbol{x}_1$, $\boldsymbol{v}_0 = \boldsymbol{0}$ and $s_0 = s_1 + 1$.
for $t = 1$ to $T$ do
    $\boldsymbol{g}_t = -\nabla J(\boldsymbol{x}_t, y)$
    $\boldsymbol{v}_t = \beta_{2,t}\boldsymbol{v}_{t-1} + (1-\beta_{2,t})\boldsymbol{g}_t^2$
    $\boldsymbol{V}_t = \text{diag}(\boldsymbol{v}_t)$
    $\hat{\boldsymbol{V}}_t = \boldsymbol{V}_t^{\frac{1}{2}} + \xi_t\boldsymbol{I}$
    $s_t = \lambda^{\frac{t}{2}}\|\boldsymbol{g}_t\|_1$
    $\beta_{1,t} = s_{t-1}/(s_t + 1)$
    $\boldsymbol{x}_{t+1} = \text{Clip}_{\boldsymbol{x}_0}^{\epsilon}\left\{\boldsymbol{x}_t - \alpha_t\hat{\boldsymbol{V}}_t^{-1}\boldsymbol{g}_t + \beta_{1,t}(\boldsymbol{x}_t - \boldsymbol{x}_{t-1})\right\}$
end for
Output: $\boldsymbol{x}_{T+1}$.
```

Assumption 1. Assume that there exists constant $G_1 > 0$ and $G_\infty > 0$ such that

$$\|\boldsymbol{g}_t\|_1 \le G_1, \|\boldsymbol{g}_t\|_\infty \le G_\infty, \forall t \ge 1.$$

Note that we do not consider the standard global smoothness assumption, i.e., the gradient Lipschitz continuity of the object function, as it is far from being satisfied in deep neural network training. Furthermore, the proof of the regret bound for AdaMSI-FGM relies on the following lemmas.

Lemma 1. *Suppose that $\forall \boldsymbol{y} \in \mathbb{R}^d$ and $\boldsymbol{x}_t \in \boldsymbol{\chi}$, then we have*

$$\langle \boldsymbol{y} - \boldsymbol{x}_t, \boldsymbol{x} - \boldsymbol{x}_t \rangle \le 0,$$

for $\forall \boldsymbol{x} \in \boldsymbol{\chi}$ if and only if $\boldsymbol{x}_t = P_{\boldsymbol{\chi}}(\boldsymbol{y})$.

The details of the proof can be found in (Bertsekas, Nedić, and Ozdaglar 2003).

Lemma 2. *Suppose that $1-\frac{1}{t} \le \beta_{2,t} \le 1-\frac{\gamma}{t}$ for some $0 < \gamma \le 1$, and $t \ge 1$, then we have*

$$\sum_{i=1}^{d}\sum_{t=1}^{T}\frac{g_{t,i}^2}{\sqrt{tV_{t,i}} + \sqrt{t}\xi_t}$$
$$\le \sum_{i=1}^{d}\frac{2(2-\gamma)}{\gamma}(\sqrt{TV_{T,i}} + \sqrt{T}\xi_T).$$

The details of the proof can be found in (Mukkamala and Hein 2017).

Theorem 1. *Let Assumption 1 , Lemma 1 and 2 hold, let $\{\boldsymbol{x}_t\}_{t=1}^T$ be generated by Equation 8. Suppose $\alpha_t = \frac{\alpha}{(s_t+1)\sqrt{t}}$, $0 < \alpha$ and $0 < \lambda < 1$, then we have the following bound on the regret:*

$$\sum_{t=1}^{T}\left[f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)\right] \le \frac{d\epsilon^2 G_\infty G_1^2}{2\alpha(1-\lambda)^2}$$
$$+\left[\frac{\epsilon^2}{2\alpha} + \frac{2\alpha(2-\gamma)}{\gamma}\right]\sum_{i=1}^{d}(\sqrt{TV_{T,i}} + \sqrt{T}\xi_T).$$

It is easy to observe that the regret bound above is mainly determined by the first term, optimizing this bound by taking $\alpha = \frac{\epsilon}{2}\sqrt{\frac{\gamma}{2-\gamma}}$. Moreover, note that for $\beta_{2,t} = 1 - \frac{1}{t}$, that is $\gamma = 1$, we recover the step-size from AdaGrad, which is beneficial for sparse gradients (Duchi, Hazan, and Singer 2010). See appendix for proof details.

Corollary 1. *Let Assumption 1 , Lemma 1 and 2 hold, let $\{\boldsymbol{x}_t\}_{t=1}^{T}$ be generated by Equation 8. Suppose $\gamma = 1$, $\alpha_t = \frac{\epsilon}{2(s_t+1)\sqrt{t}}$, $\xi_t = \frac{\delta}{\sqrt{t}}$, $\delta > 0$ and $0 < \lambda < 1$, then we have the following bound on the regret:*

$$\sum_{t=1}^{T}[f(\boldsymbol{x}_t) - f(\boldsymbol{x}^*)] \le 2\epsilon\sum_{i=1}^{d}(\|\boldsymbol{g}_{1:T,i}\|_2 + \delta) + \frac{d\epsilon G_\infty G_1^2}{(1-\lambda)^2}.$$

The above corollary implies that AdaMSI-FGM obtains an $O(\sum_{i=1}^{d}\|\boldsymbol{g}_{1:T,i}\|_2)$ regret bound, which is $O(\sqrt{T})$ in the worst case. From this regret bound, we gain several insights:

- If $\lambda = 0$, momentum will be removed from AdaMSI-FGM, which does not affect the essence of the regret bound in the worst case. However, whether the optimal lower bound will be damaged remains an open problem. On the other hand, the value choice of λ should not be too close to 1 for a tighter regret bound, otherwise the second term of regret bound may grow well beyond $O(\sqrt{T})$.
- Due to the use of adaptive step-size, the regret bound of AdaMSI-FGM is data-dependent and becomes tighter whenever the gradients are small or sparse such that $\|\boldsymbol{g}_{1:T,i}\|_2 \ll G_\infty\sqrt{T}$. In addition, this bound can be easily translated into a data-dependent $O(\frac{1}{\sqrt{T}})$ convergence rate for stochastic convex optimization using the online-to-batch conversion (Kakade and Tewari 2008).
- The derived regret bound eliminates the logarithmic factor present in the optimal $O(\sqrt{T})$ regret bound and the $O(\sqrt{\log(T)T})$ bound obtained by traditional Adam-type methods. The logarithmic factor arises when bounding $\sum_{t=1}^{T}\alpha_t\|\boldsymbol{m}_t\|^2_{\hat{\mathbf{V}}_t^{-1}}$ in Adam variants, which does not occur in AdaMSI-FGM since it uses HB momentum instead of FME momentum with the $\boldsymbol{m}_t$ term.

Experiments

We conduct extensive experiments on the ImageNet dataset to validate the effectiveness of the proposed methods.

Experimental Setting

Dataset. We randomly select 500 images from ILSVRC 2012 validation set.

Models. We consider eight pre-trained models from the torchvision library (Paszke et al. 2019) on ImageNet dataset. These models include ResNet34 (ResNet) (He et al. 2016a), EfficientNet-b0 (EfficientNet) (Tan and Le 2019), GoogLeNet (GoogLeNet) (Szegedy et al. 2015), MNASNet-0-5 (MNASNet) (Tan et al. 2019), MobileNet-v3-small (MobileNet) (Howard et al. 2019), ShuffleNet-v2-x0-5 (ShuffleNet) (Ma et al. 2018), SqueezeNet-1-1 (SqueezeNet) (Iandola et al. 2016) and VGG11 (VGG) (Simonyan and Zisserman 2015). All eight models are used as both source models for generating adversarial examples and target models for testing these adversarial examples.

Baselines. We select the popular adversarial attack methods PGD (Madry et al. 2018), AutoAttack (Croce and Hein 2020), MI-FGSM (Dong et al. 2018) and NI-FGSM (Lin et al. 2020) as our baselines. Our focus is on transfer-based attacks, where no query access to the target model is granted. Therefore, we do not compare with query-based methods.

Infrastructure. The experiments are conducted on a single NVIDIA GeForce RTX 3060 GPU. Some experiments follow (Kim 2020) to support the state-of-the-art baselines. The software versions used are Ubuntu 18.04.1, Python 3.7.12, PyTorch 1.11.0, and Torchvision 0.12.0.

Hyper-Parameters. Although more iterations are favorable to convergence (Pintor et al. 2022) and targeted attacks (Zhao, Liu, and Larson 2021), the traditional iteration setting $T = 10$ is adopted since we focus on non-targeted transferable attacks. The maximum of L_∞ norm perturbation $\epsilon = 4/255$, and the batch-size is set to 64 for all algorithms. For MI-FGSM and NI-FGSM, we adopt the default momentum parameter $\mu = 1$ and step-size $\alpha_T = 4/255/10$. For PGD, the step-size $\alpha_T = 4/255/10$. For AdaMSI-FGM, we set $\alpha_t \equiv 1/255/10$, $\lambda = 0.6$, $\beta_{2,t} = 1 - \frac{\gamma}{t}$ where $\gamma = 1$, and $\xi_t = \frac{\delta}{\sqrt{t}}$ where $\delta = 1e - 16$.

Metric. The standard evaluation metric for adversarial attacks, Attack Success Rate (ASR), is used to assess the quality of the adversarial example. Higher ASR values indicate better adversarial example quality.

Results of Adversarial Attacks

Comparison with Classic Attacks. The attack success rates against the considered models are presented in Table 1. From the table, we observe that all algorithms exhibit strong white-box attack performance, achieving nearly 100% ASRs against all white-box models. By integrating HB momentum with parameter adaption and adaptive step-size scheme, our AdaMSI-FGM outperforms both PGD and AutoAttack in black-box attacks. Notably, our method consistently achieves $2\% \sim 17.8\%$ higher ASRs than PGD and AutoAttack under black-box attack cases, demonstrating the effectiveness of the proposed algorithm.

Comparison with Momentum Attacks. Our algorithm is improved from MI-FGSM to bridge the theory-practice gap and guarantee convergence, therefore, we mainly compare it to MI-FGSM and follow the experiment presented in (Dong et al. 2018). Seven models are introduced into the experiment which are Inception v3 (Inc-v3) (Szegedy et al. 2016), Inception v4 (Inc-v4), Inception Resnet v2 (IncRes-v2) (Szegedy et al. 2017), Resnet v2-101 (Res-101) (He et al. 2016b) and the other three of which are adversarially trained models—Inc-v3ens3, Inc-v3ens4, IncRes-v2ens (Tramèr et al. 2018). The results are shown in Table 2, it can be seen that our algorithm has stronger attack performance against the adversarially trained model than MI-FGSM. Besides, we also test MI-FGSM, NI-FGSM and AdaMSI-FGM with the eight normally trained models used in Table 1. We

Model	Attack	ResNet	EfficientNet	GoogLeNet	MNASNet	MobileNet	ShuffleNet	SqueezeNet	VGG
ResNet	PGD	**100.0***	36.6	40.4	43.4	40.2	50.6	52.6	46.6
	AutoAttack	**100.0***	34.6	40.2	42.8	40.8	50.0	54.0	48.4
	Ours	**100.0***	**51.2**	**55.4**	**54.6**	**46.8**	**55.2**	**65.0**	**62.4**
EfficientNet	PGD	42.8	99.4*	40.8	54.4	48.0	51.0	53.4	49.2
	AutoAttack	41.8	**100.0***	40.0	53.2	48.0	51.2	53.8	49.2
	Ours	**58.0**	99.0*	**53.2**	**67.0**	**59.4**	**60.8**	**63.2**	**61.4**
GoogLeNet	PGD	38.8	33.8	**100.0***	39.6	41.2	48.6	49.8	43.2
	AutoAttack	38.8	33.6	**100.0***	41.6	41.2	49.2	53.8	44.8
	Ours	**50.0**	**42.8**	**100.0***	**49.0**	**45.4**	**53.6**	**60.2**	**54.2**
MNASNet	PGD	37.0	36.2	35.4	**100.0***	45.0	49.8	51.8	42.2
	AutoAttack	35.2	32.4	34.4	**100.0***	45.8	49.6	50.6	41.2
	Ours	**44.6**	**46.2**	**42.4**	**100.0***	**57.8**	**56.6**	**60.8**	**52.2**
MobileNet	PGD	35.6	36.6	35.6	53.4	**100.0***	50.4	51.8	42.0
	AutoAttack	33.6	32.6	35.0	51.6	**100.0***	50.8	49.8	40.8
	Ours	**43.0**	**48.0**	**43.4**	**67.0**	**100.0***	**56.4**	**59.4**	**51.0**
ShuffleNet	PGD	34.0	31.6	34.2	42.2	40.4	**100.0***	51.0	39.6
	AutoAttack	32.4	28.0	34.8	40.4	39.0	**100.0***	48.8	38.6
	Ours	**38.6**	**35.4**	**37.8**	**50.2**	**44.6**	**100.0***	**58.6**	**44.4**
SqueezeNet	PGD	37.0	31.8	36.6	43.6	40.4	51.2	**100.0***	46.4
	AutoAttack	33.6	29.8	35.4	40.8	38.2	49.2	**100.0***	42.2
	Ours	**43.2**	**37.8**	**44.4**	**50.6**	**45.8**	**56.4**	**100.0***	**54.8**
VGG	PGD	43.8	38.4	39.2	46.8	43.0	49.4	59.0	**100.0***
	AutoAttack	42.4	37.0	39.8	46.0	41.6	48.6	58.6	**100.0***
	Ours	**58.2**	**53.2**	**52.2**	**58.6**	**48.6**	**55.2**	**70.0**	**100.0***

Table 1: Attack success rates (%) of adversarial attacks against eight models. * indicates the white-box attacks.

have included the experimental results in supplementary materials due to page limitations. Our algorithm has improved in black-box ASRs compared to MI-FGSM and NI-FGSM, which indicates that it is successful in replacing the sign function in the momentum attack algorithms with an adaptive step-size strategy.

Further Analysis

Convergence Comparison. To compare the convergence between this method and existing methods, we use BIM and AdaMSI-FGM to generate the adversarial examples on ResNet34. As shown in Figure 1, BIM achieving a stationary value (but being unable to reduce the score any more), which is consistent with the point that sign-based method cannot converge to optimal value. The AdaMSI-FGM curve declinating to a lower value, indicating our approach has the impact of accelerating convergence.

Flexibility. The proposed method can be combined with other existing black-box attack methods such as DI-FGSM (DI) (Xie et al. 2019) and TI-FGSM (TI) (Dong et al. 2019). The experimental results of algorithm DI+Ours obtained by combining DI and AdaMSI-FGM are presented in Table 3. We observe that the integration of the two algorithms significantly improve transferability, demonstrating the effectiveness and flexibility of AdaMSI-FGM. More experimental results of other composite methods can be found in supplementary materials.

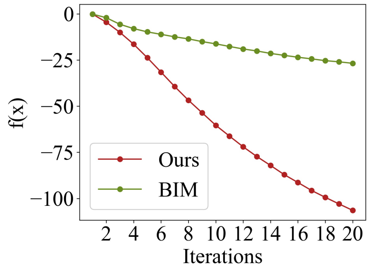


Figure 1: Values of loss vs. iterations. $f(\boldsymbol{x}) = -J(\boldsymbol{x}, y)$.

Human-Imperceptible. We visualize nine adversarial examples generated by PGD, MI-FGSM and AdaMSI-FGM. The original images are shown in Figure 2(a). We choose ResNet34 as source model and Inception v3 as target model. The resulting adversarial examples are displayed in Figure 2(b), 2(c), and 2(d). It is noteworthy that all of these adversarial noises are human-imperceptible.

Discussion

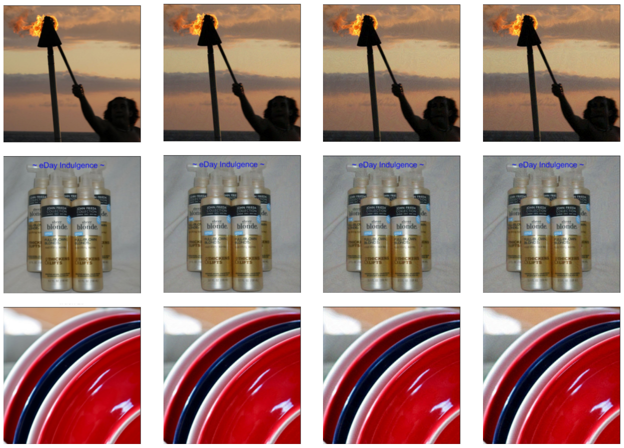
The generation of adversarial examples using AdaMSI-FGM can be seen as analogous to training neural network models with adaptive SGDM, thus the transferability of our method is inherited from its generalizability in model training. Furthermore, the adaptive step-size motivates us to automatically assign different learning rates to each dimension

Model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{\mathrm{ens3}}$	Inc-v3$_{\mathrm{ens4}}$	IncRes-v2$_{\mathrm{ens}}$
Inc-v3	MI-FGSM	**100.0***	44.5	42.9	35.4	13.7	12.8	6.4
	Ours	99.3*	**47.9**	**43.6**	**37.5**	**15.1**	**13.2**	**6.8**
Inc-v4	MI-FGSM	54.9	**99.8***	45.8	**40.7**	16.8	14.8	7.3
	Ours	**56.1**	99.2*	**46.9**	**40.7**	**19.2**	**16.7**	**9.3**
IncRes-v2	MI-FGSM	**59.2**	**49.5**	**97.9***	**45.1**	22.9	16.3	11.0
	Ours	53.1	45.7	96.1*	40.9	**25.0**	**18.6**	**16.4**
Res-101	MI-FGSM	**57.9**	51.4	**49.4**	**99.3***	24.2	21.3	12.0
	Ours	56.4	**52.1**	47.2	96.5*	**27.7**	**23.5**	**14.7**

Table 2: Attack success rates (%) of adversarial attacks against seven models. * indicates the white-box attacks.

Model	Attack	ResNet	EfficientNet	GoogLeNet	MNASNet	MobileNet	ShuffleNet	SqueezeNet	VGG
ResNet	DI	**100.0***	44.2	53.2	49.8	45.2	53.6	60.0	54.8
	DI+Ours	**100.0***	**57.8**	**61.8**	**59.4**	**48.8**	**58.4**	**67.6**	**66.2**
EfficientNet	DI	54.2	**98.8***	50.6	62.4	58.6	56.2	60.4	58.4
	DI+Ours	**61.2**	**98.8***	**62.2**	**70.8**	**66.8**	**63.2**	**67.0**	**64.6**
GoogLeNet	DI	46.8	39.6	**100.0***	47.6	45.0	50.8	56.0	50.6
	DI+Ours	**57.6**	**48.2**	**100.0***	**55.8**	**48.2**	**57.8**	**62.4**	**59.2**
MNASNet	DI	42.4	42.0	40.0	99.8*	54.4	53.2	56.8	47.8
	DI+Ours	**48.4**	**50.6**	**48.0**	**100.0***	**63.0**	**59.2**	**64.2**	**58.0**
MobileNet	DI	38.6	43.8	40.6	61.8	**100.0***	54.0	56.0	47.8
	DI+Ours	**45.2**	**51.4**	**47.0**	**73.4**	**100.0***	**61.2**	**63.6**	**55.8**
ShuffleNet	DI	37.8	34.4	38.0	47.4	44.6	**100.0***	56.2	42.2
	DI+Ours	**39.4**	**40.2**	**40.2**	**53.4**	**45.8**	**100.0***	**63.4**	**47.6**
SqueezeNet	DI	41.0	35.8	44.0	50.4	44.6	55.0	**100.0***	52.2
	DI+Ours	**46.2**	**40.6**	**49.2**	**58.0**	**46.6**	**61.6**	**100.0***	**59.6**
VGG	DI	53.8	47.8	50.8	55.8	47.8	51.6	65.6	**100.0***
	DI+Ours	**62.8**	**56.8**	**60.6**	**65.4**	**51.6**	**58.8**	**74.0**	**100.0***

Table 3: Attack success rates (%) of adversarial attacks against eight models. * indicates the white-box attacks.



(a) Original (b) PGD (c) MI-FGSM (d) Ours

Figure 2: Original images vs. adversarial examples.

of the adversarial noise, which also help our method. Other superior algorithms in optimization can be used to boost the adversarial attacks performance, but it needs to be discussed whether they can avoid the potential threat of the sign function and ensure convergence

Conclusion

We propose a novel adversarial attack method that guarantees optimal convergence under milder assumptions in general convex settings. Specifically, we do not assume the monotonicity of the momentum parameter or the smoothness of the objective function. Instead, we adaptively adjust the momentum and step-size parameters based solely on gradient information, which not only bridges the theory-practice gap but also avoids potential issues associated with the sign function. Under these realistic assumptions, we obtain a data-dependent $O(\sqrt{T})$ regret bound, eliminating the logarithmic factor typically present in Adam-type methods. Our method successfully generates adversarial examples with human-imperceptible noises while achieving high attack success rates, demonstrating its superiority.

Acknowledgments

This work was supported in part by National Natural Science Foundation of China (No. 62101571 and No. 62106281).

References

Alacaoglu, A.; Malitsky, Y.; Mertikopoulos, P.; and Cevher, V. 2020. A new regret analysis for Adam-type algorithms. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, 202–210. PMLR.

Balles, L.; and Hennig, P. 2018. Dissecting Adam: The Sign, Magnitude and Variance of Stochastic Gradients. In *ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, 413–422. PMLR.

Bernstein, J.; Wang, Y.; Azizzadenesheli, K.; and Anandkumar, A. 2018. SIGNSGD: Compressed Optimisation for Non-Convex Problems. In *ICML 2018*, volume 80 of *Proceedings of Machine Learning Research*, 559–568. PMLR.

Bertsekas, D. P.; Nedić, A.; and Ozdaglar, A. E. 2003. Convex Analysis and Optimization.

Cai, Y.; Hu, X.; Wang, H.; Zhang, Y.; Pfister, H.; and Wei, D. 2021. Learning to Generate Realistic Noisy Images via Pixel-level Noise-aware Adversarial Training. In *NeurIPS 2021*, 3259–3270.

Chen, X.; Liang, C.; Huang, D.; Real, E.; Wang, K.; Liu, Y.; Pham, H.; Dong, X.; Luong, T.; Hsieh, C.; Lu, Y.; and Le, Q. V. 2023. Symbolic Discovery of Optimization Algorithms. *CoRR*, abs/2302.06675.

Crawshaw, M.; Liu, M.; Orabona, F.; Zhang, W.; and Zhuang, Z. 2022. Robustness to Unbounded Smoothness of Generalized SignSGD. In *NeurIPS 2022*.

Croce, F.; and Hein, M. 2020. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML 2020*, volume 119 of *Proceedings of Machine Learning Research*, 2206–2216. PMLR.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting Adversarial Attacks With Momentum. In *CVPR 2018*, 9185–9193. Computer Vision Foundation / IEEE Computer Society.

Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks. In *CVPR 2019*, 4312–4321. Computer Vision Foundation / IEEE.

Duchi, J. C.; Hazan, E.; and Singer, Y. 2010. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. In *COLT 2010*, 257–269. Omnipress.

Gao, L.; Zhang, Q.; Zhu, X.; Song, J.; and Shen, H. T. 2021. Staircase Sign Method for Boosting Adversarial Attacks. *CoRR*, abs/2104.09722.

Ghadimi, E.; Feyzmahdavian, H. R.; and Johansson, M. 2015. Global convergence of the Heavy-ball method for convex optimization. In *14th European Control Conference, ECC 2015*, 310–315. IEEE.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and Harnessing Adversarial Examples. In *ICLR 2015*.

Goyal, P.; Dollár, P.; Girshick, R. B.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; and He, K. 2017. Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour. *CoRR*, abs/1706.02677.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep Residual Learning for Image Recognition. In *CVPR 2016*, 770–778. IEEE Computer Society.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity Mappings in Deep Residual Networks. In *ECCV 2016*, volume 9908 of *Lecture Notes in Computer Science*, 630–645. Springer.

Howard, A.; Pang, R.; Adam, H.; Le, Q. V.; Sandler, M.; Chen, B.; Wang, W.; Chen, L.; Tan, M.; Chu, G.; Vasudevan, V.; and Zhu, Y. 2019. Searching for MobileNetV3. In *ICCV 2019*, 1314–1324. IEEE.

Iandola, F. N.; Moskewicz, M. W.; Ashraf, K.; Han, S.; Dally, W. J.; and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1MB model size. *CoRR*, abs/1602.07360.

Kakade, S. M.; and Tewari, A. 2008. On the Generalization Ability of Online Strongly Convex Programming Algorithms. In *NeurIPS, 2008*, 801–808. Curran Associates, Inc.

Karimireddy, S. P.; Rebjock, Q.; Stich, S. U.; and Jaggi, M. 2019. Error Feedback Fixes SignSGD and other Gradient Compression Schemes. In *ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, 3252–3261. PMLR.

Kim, H. 2020. Torchattacks : A Pytorch Repository for Adversarial Attacks. *CoRR*, abs/2010.01950.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *ICLR 2015*.

Kunstner, F.; Chen, J.; Lavington, J. W.; and Schmidt, M. 2023. Noise Is Not the Main Factor Behind the Gap Between Sgd and Adam on Transformers, But Sign Descent Might Be. In *ICLR 2023*. OpenReview.net.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2017. Adversarial examples in the physical world. In *ICLR 2017*. OpenReview.net.

Li, X.; Liu, M.; and Orabona, F. 2022. On the Last Iterate Convergence of Momentum Methods. In *International Conference on Algorithmic Learning Theory*, volume 167 of *Proceedings of Machine Learning Research*, 699–717. PMLR.

Lin, J.; Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2020. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. In *ICLR 2020*. OpenReview.net.

Liu, S.; Chen, P.; Chen, X.; and Hong, M. 2019. signSGD via Zeroth-Order Oracle. In *ICLR 2019*. OpenReview.net.

Ma, N.; Zhang, X.; Zheng, H.; and Sun, J. 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. In *ECCV 2018*, volume 11218 of *Lecture Notes in Computer Science*, 122–138. Springer.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *ICLR 2018*. OpenReview.net.

Mukkamala, M. C.; and Hein, M. 2017. Variants of RMSProp and Adagrad with Logarithmic Regret Bounds. In *ICML 2017*, volume 70 of *Proceedings of Machine Learning Research*, 2545–2553. PMLR.

Nesterov, Y. 1983. A method for solving the convex programming problem with convergence rate $O(1/k^2)$. *Proceedings of the USSR Academy of Sciences*, 269: 543–547.

Ochs, P.; Chen, Y.; Brox, T.; and Pock, T. 2014. iPiano: Inertial Proximal Algorithm for Nonconvex Optimization. *SIAM J. Imaging Sci.*, 7(2): 1388–1419.

Pang, T.; Yang, X.; Dong, Y.; Xu, T.; Zhu, J.; and Su, H. 2020. Boosting Adversarial Training with Hypersphere Embedding. In *NeurIPS 2020*.

Papernot, N.; McDaniel, P. D.; and Goodfellow, I. J. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. *CoRR*, abs/1605.07277.

Papernot, N.; McDaniel, P. D.; Goodfellow, I. J.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical Black-Box Attacks against Machine Learning. In *AsiaCCS 2017*, 506–519. ACM.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; Desmaison, A.; Köpf, A.; Yang, E. Z.; DeVito, Z.; Raison, M.; Tejani, A.; Chilamkurthy, S.; Steiner, B.; Fang, L.; Bai, J.; and Chintala, S. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *NeurIPS 2019*, 8024–8035.

Pintor, M.; Demetrio, L.; Sotgiu, A.; Demontis, A.; Carlini, N.; Biggio, B.; and Roli, F. 2022. Indicators of Attack Failure: Debugging and Improving Optimization of Adversarial Examples. In *NeurIPS 2022*.

Polyak, B. 1964. Some methods of speeding up the convergence of iteration methods. *Ussr Computational Mathematics and Mathematical Physics*, 4: 1–17.

Reddi, S. J.; Kale, S.; and Kumar, S. 2018. On the Convergence of Adam and Beyond. In *ICLR 2018*. OpenReview.net.

Ruder, S. 2016. An overview of gradient descent optimization algorithms. *CoRR*, abs/1609.04747.

Shafahi, A.; Najibi, M.; Ghiasi, A.; Xu, Z.; Dickerson, J. P.; Studer, C.; Davis, L. S.; Taylor, G.; and Goldstein, T. 2019. Adversarial training for free! In *NeurIPS 2019*, 3353–3364.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015*.

Sun, T.; Li, D.; Quan, Z.; Jiang, H.; Li, S.; and Dou, Y. 2019. Heavy-ball Algorithms Always Escape Saddle Points. In *IJCAI 2019*, 3520–3526. ijcai.org.

Sutskever, I.; Martens, J.; Dahl, G. E.; and Hinton, G. E. 2013. On the importance of initialization and momentum in deep learning. In *ICML 2013*, volume 28 of *JMLR Workshop and Conference Proceedings*, 1139–1147. JMLR.org.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In *AAAI 2017*, 4278–4284. AAAI Press.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S. E.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR 2015*, 1–9. IEEE Computer Society.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the Inception Architecture for Computer Vision. In *CVPR 2016*, 2818–2826. IEEE Computer Society.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR 2014*.

Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; and Le, Q. V. 2019. MnasNet: Platform-Aware Neural Architecture Search for Mobile. In *CVPR 2019*, 2820–2828. Computer Vision Foundation / IEEE.

Tan, M.; and Le, Q. V. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In *ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, 6105–6114. PMLR.

Tao, W.; Bao, L.; Long, S.; Wu, G.; and Tao, Q. 2023. Adapting Step-size: A Unified Perspective to Analyze and Improve Gradient-based Methods for Adversarial Attacks. *CoRR*, abs/2301.11546.

Tao, W.; Long, S.; Wu, G.; and Tao, Q. 2021. The Role of Momentum Parameters in the Optimal Convergence of Adaptive Polyak's Heavy-ball Methods. In *ICLR 2021*. OpenReview.net.

Tao, W.; Wu, G.; and Tao, Q. 2022. Momentum Acceleration in the Individual Convergence of Nonsmooth Convex Optimization With Constraints. *IEEE Trans. Neural Networks Learn. Syst.*, 33(3): 1107–1118.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I. J.; Boneh, D.; and McDaniel, P. D. 2018. Ensemble Adversarial Training: Attacks and Defenses. In *ICLR 2018*. OpenReview.net.

Wang, G.; Lu, S.; Cheng, Q.; Tu, W.; and Zhang, L. 2020. SAdam: A Variant of Adam for Strongly Convex Functions. In *ICLR 2020*. OpenReview.net.

Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; and Yuille, A. L. 2019. Improving Transferability of Adversarial Examples With Input Diversity. In *CVPR 2019*, 2730–2739. Computer Vision Foundation / IEEE.

Yang, T.; Lin, Q.; and Li, Z. 2016. Unified Convergence Analysis of Stochastic Momentum Methods for Convex and Non-convex Optimization. *arXiv: Optimization and Control*.

Zhao, Z.; Liu, Z.; and Larson, M. A. 2021. On Success and Simplicity: A Second Look at Transferable Targeted Attacks. In *NeurIPS 2021*, 6115–6128.

Zhuang, J.; Tang, T.; Ding, Y.; Tatikonda, S.; Dvornek, N. C.; Papademetris, X.; and Duncan, J. S. 2020. AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients. In *NeurIPS 2020*.