IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 14 August 2025, accepted 7 September 2025,
date of publication 10 September 2025, date of current version 16 September 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3608117

RESEARCH ARTICLE

Game Theoretic Mixed Experts for Combinational Adversarial Machine Learning

KALEEL MAHMOOD[1], ETHAN RATHBUN[2], RONAK SAHU[3], MARTEN VAN DIJK[4], (Fellow, IEEE), SOHAIB AHMAD[5], AND CAIWEN DING[6], (Member, IEEE)

[1]Department of Computer Science and Statistics, The University of Rhode Island, Kingston, RI 02881, USA
[2]Khoury College of Computer Sciences, Northeastern University, Boston, MA 02115, USA
[3]School of Computing, University of Connecticut, Storrs, CT 06269, USA
[4]CWI Amsterdam, 1098 XG Amsterdam, The Netherlands
[5]Visa Inc., San Francisco, CA 94105, USA
[6]Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, USA

Corresponding author: Kaleel Mahmood (kaleel.mahmood@uri.edu)

ABSTRACT Recent advances in adversarial machine learning have shown that defenses previously considered robust are actually susceptible to adversarial attacks which are specifically customized to target their weaknesses. However, whether the adversarial examples generated by customized attacks, are effective on other defenses, is an open question. In this work we seek to explore three important security questions: First, do different defense strategies exhibit the same low transferability properties as different model architectures and, if so, how can this low transferability be utilized to improve robustness? Second, how can a white-box adversary design attacks to specifically thwart multi-defense based setups? Last, how can game theoretic analysis further improve the robustness against an adversary capable of implementing multiple state-of-the-art attacks? To this end we provide multiple contributions, including the first transferability study between multiple defense strategies, three new attack algorithms designed to break random transform and ensemble defenses, and two game theoretic frameworks for analyzing and optimizing robustness over a combination of adversarial attacks and defenses. Empirically, we show our framework is 18% more robust on CIFAR-10 and is 27% more robust on Tiny-ImageNet than the best single state-of-the-art defense that we analzye.

INDEX TERMS Adversarial machine learning, adversarial examples, adversarial defense, deep learning.

I. INTRODUCTION

Machine learning models have been shown to be vulnerable to adversarial examples (AEs) [15], [33], [50]. AEs are inputs with small perturbations added, such that machine learning models misclassify them with high confidence. Addressing the security risks posed by AEs are critical for the safe deployment of machine learning in areas like health care [14] and self driving vehicles [35]. Current defenses and attacks in adversarial machine learning have trended towards a cat and mouse dynamic, where new defenses are being proposed and then broken [6], [26], [42], [44] by improved attacks. Many of these attacks are *specialized* to exploit particular vulnerabilities of specific defense strategies.

Some works [2], [29], [40] have looked at combinations of defenses as a solution to this dynamic. However, these works have shown limited improvements in robustness and fail to incorporate diverse set of defense and state-of-the-art strategies such as random transforms [36], [48] or diffusion based adversarial training [47]. They also do not consider the impact of adaptive attacks nor do they explore the optimization of their methods to defend against them.

In parallel to attack and defense development, studies have also been conducted on the transferability of AEs [24], [27], [49]. Transferabiltiy refers to the phenomena where AEs generated for one model are also misclassified by a different machine learning model. However, to the best of our knowledge, no analyses has been done on the relationship between transferability, defense strategies, and specialized attacks. Low transferability between defense strategies may

The associate editor coordinating the review of this manuscript and approving it for publication was SK Hafizul Islam.

point towards further improvements in robustness found in combinational voting strategies. Thus, several pertinent questions arise:

1) *Do AEs generated for one specific defense transfer to other defenses?*
2) *Can low transferability between defense strategies be exploited in vanilla or detector based combination?*
3) *Can adversarial attacks be designed to target multiple different defense paradigms simultaneously?*
4) *Can a substantial gain in robustness be achieved through the utilization of game-theoretic analyses?*

These are precisely the questions our paper seeks to answer. We break from the traditional dynamic of adversarial machine learning, which focuses on the single best attack and defense. We instead take a multi-faceted approach and use a mixture of state-of-the-art attacks and defenses to answer the above questions. We provide the following contributions:

1) **Defense Transferability Analysis** - We analyze the adversarial transferability of state-of-the-art defenses like Trash is Treasure [48], Barrage of Random Transforms [36], Friendly Adversarial Training [52], diffusion based Wide Resnets [47], and other new architectures like SNNs [13], [37] and ViTs [12]. We show specific attacks on defenses do not transfer well. This means there is potential for a combinational defense within a game theoretic framework.
2) **New Adversarial Attacks** - We develop three new attacks, the Momentum Iterative Method over Expectation attack (MIME), the Multi-Agent Gradient Expectation attack (MAGE) and detector based Multi-Agent Gradient Expectation attack (MAGE-D). These attacks are designed to create the strongest possible adversary to attack randomized and multi-classifier defenses.
3) **Game theoretic Mixed Experts** - We propose a game-theoretic framework for finding approximately optimal strategies for adversarial attackers and defenders that can implement multi-model attack and defense techniques. We mathematically derive our framework, and empirical test it by implementing multiple state-of-the-art adversarial attacks and defenses on two datasets, CIFAR-10 [21] and Tiny ImageNet [23]. Through our framework we demonstrate that we are able to achieve a 18% increase in robustness on CIFAR-10, and a 27% increase on Tiny ImageNet over all single-model defenses we tested when evaluating on non-adaptive attacks.
4) **Adaptive Game theoretic Mixed Experts** - Motivated by developments in adaptive attacks such as AutoAttack [9], we further build upon the GaME framework by proposing Ada-GAME. With this framework we are able to demonstrate an improvement in robust accuracy over all other ensemble baselines for both datasets.

A. PAPER OVERVIEW

All the subjects previously mentioned are interlinked and studied deeply in this paper. We begin in Section II by introducing and briefly explaining the mechanics behind each defense strategy we study in this paper. In Section III we discuss the adversarial threat model for this work. In Section IV we develop specialized attacks to adapt to random transform, multi-model, and detector based defenses techniques. Following the development of these attacks, we study the transferability adversarial examples between each of the defenses on the CIFAR-10 [21] dataset in Section V. These results display surprisingly low levels of transferability between defense strategies, indicating further robustness gains to be obtained from multi-model defense approaches. Motivated by this, we develop two new game-theoretic frameworks for optimizing mixtures of adversarial defenses, GaME and Ada-GaME, in Section VI. Both these frameworks allow for the optimization of any mixture of pre-trained defense or vanilla models against adversarial attacks without further model training. In Section VII we present extensive experimental results testing the capabilities of the GaME and Ada-GaME frameworks, displaying increased robustness over all single-model or uniform defenses strategies against both adaptive and non-adaptive attacks. We then offer concluding remarks in Section VIII.

Overall, our paper advances the field of adversarial machine learning by showing the potential for combinational defense approaches in a theoretical and principled manner that extends beyond the conventional single attack, single defense standard.

II. ADVERSARIAL MACHINE LEARNING DEFENSES

In this section we give an overview of the diverse set of defenses we study in this paper. Our analyses first encompasses defenses based on randomization, adversarial training, and exploiting model transferability. This includes the Barrage of Random Transforms (BART), Friendly Adversarial Training (FAT), Trash is Treasure (TiT), and diffusion based Wide Resnet defenses respectively. We additionally consider multiple model architectures including Big Transfer models (BiTs), Vision Transformers (ViTs) and Spiking Neural Networks (SNNs).

Despite our broad range, we do not attempt to test every novel adversarial defense. It is simply infeasible to test every proposed adversarial machine learning defense, as new defenses are constantly being produced. However, based on our game theoretic design and open source code (which will be provided upon publication), any new defense can easily be tested and integrated into our framework.

A. BARRAGE OF RANDOM TRANSFORMS

Barrage of Random Transforms (BaRT) [36] utilize a set of image transformations in a random order and with randomized transformation parameters to thwart adversarial attacks. In this paper, we work with the original BaRT implementation which includes both differentiable and non-differentiable image transformations.

Let $t_j^i(x)$ represent the i^{th} transformation used in the j^{th} order in the sequence. A BaRT defense using n image

transformations randomly alters the input x:

$$t(x) = t_{\mu_n}^{\omega_n} \circ t_{\mu_{n-1}}^{\omega_{n-1}} \circ \ldots \circ \quad (1)$$

where ω represents the subset of n transformations randomly selected from a set of N total possible transformations and μ represents the randomized order in which the n transformations are applied. In Equation 1 the parameters of each image transformation $t_j^i(x)$ are also randomized at run time, further adding to the stochastic nature of the defense.

Why we selected it: Many defenses are broken soon after being proposed [44]. BaRT is one of the few defenses that has continued to show robustness even when attacks are specifically tailored to work against it. For example, most recently BaRT achieves 29% robustness on CIFAR-10 against a customized white-box attack [42]. It remains an open question whether using BaRT with other randomized approaches (i.e. selecting between different defenses) can yield even greater robustness.

B. FRIENDLY ADVERSARIAL TRAINING

Training classifiers to correctly recognize adversarial examples was originally proposed in [15] using FGSM. This concept was later expanded to include training on adversarial examples generated by PGD in [25]. In [52], it was shown that Friendly Adversarial Training (FAT) could achieve high clean accuracy, while maintaining robustness to adversarial examples. This training was accomplished by using a modified version of PGD called PGD-K-τ. In PGD-K-τ, K refers to the number of iterations used for PGD. The τ variable is a hyperparameter used in training which stops the PGD generation of adversarial examples earlier than the normal K number of steps, if the sample is already misclassified.

Why we selected it: There are many different defenses that rely on adversarial training [25], [28], [46], [51] and training and testing them all is not computationally feasible. We selected FAT for its good trade off between clean accuracy and robustness, and because we wanted to test adversarial training on both Vision Transformer and CNN models. In this regard, FAT is one of the adversarial training methods that has already been demonstrated to work across both types of architectures [27].

C. TRASH IS TREASURE

One early direction in adversarial defense design was model ensembles [32]. However, due to the high transferability of adversarial examples between models, such defenses were shown to not be robust [44]. Trash is Treasure (TiT) [48] is a two model defense that seeks to overcome the transferability issue by training one model $C_a(\cdot)$ on the adversarial examples from another model $C_b(\cdot)$. At run time both models are used:

$$y = C_a(\psi(x, C_b)) \quad (2)$$

where ψ is an attack done on model C_b with input x and C_a is the classifier that makes the final class label prediction on the adversarial example generated by ψ with C_b.

Why we selected it: TiT is one of the newest defenses that tries to achieve robustness in a way that is fundamentally different than pure randomization strategies or direct adversarial training. In our paper, we further develop two versions of TiT. One version is based on the original proposed CNN-CNN implementation. We also test a second mixed architecture version using Big Transfer model and Vision Transformers to try and leverage the low transferability phenomena described in [27].

D. DIFFUSION BASED CLASSIFIERS

Many works have observed that increased latent space coverage of the training dataset can further improve the robustness of machine learning classifiers against adversarial attacks [38], [43]. However, the gathering of further, external data is costly and potentially subject to poisoning attacks, requiring further dataset auditing and cost. Therefore, many works [39], [45] have aimed to utilize diffusion based generative models [18] to further fill the gaps in established datasets and improve the robustness of classification models against adversarial attacks. In particular, in this work we use the diffusion based Wide Resnet models (WRN-28-10) trained in [47] on the CIFAR-10 and Tiny ImageNet datasets.

Why we selected it: Classifiers trained with diffusion based adversarial training have recently been shown to be the top performers in terms of both robust accuracy and clean accuracy on standard datasets like CIFAR-10. While these models typically require longer training times to converge, the benefits they offer are immense, especially when they are readily provided online, as in the case of [47]. As these are the strongest single models in our study, they represent the state-of-the-art benchmark for robustness that we want to further build upon in combinational approaches.

E. NOVEL ARCHITECTURES

In addition to adversarial machine learning defenses, we also include several novel architectures that have recently achieved state-of-the-art or near state-of-the-art performance in image recognition tasks. These include the Vision Transformer (ViT) [12] and Big Transfer models (BiT) [19]. Both of these types of models utilize pre-training on larger datasets and fine tuning on smaller datasets to achieve high fidelity results. We also test Spiking Neural Network (SNNs) architectures. SNNs are a competitor to artificial neural networks that can be described as a linear time invariant system with a network structure that employs non-differentiable activation functions [49]. A major challenge in SNNs has been matching the depth and model complexity of traditional deep learning models. Two approaches have been used to overcome this challenge, the Spiking Element Wise (SEW) ResNet [13] and transferring weights from existing CNN architectures to SNNs [37]. We experiment with both approaches in our paper.

Why we selected it: The set of adversarial examples used to attack one type of architecture (e.g. a ViT) have shown to not be misclassified by other architecture types (e.g. a BiT or

SNN) [27], [49]. While certain white-box attacks have been used to break multiple undefended models, it remains an open question if different architectures combined with different defenses can yield better performance.

III. ADVERSARIAL THREAT MODEL

We assume a white-box adversarial threat model. This means the attacker is aware of the set of all defenses D that the defender may use for prediction. In addition, $\forall d \in D$ the attacker also knows the classifier weights θ_d, architecture and any input image transformations the defense may apply. To generate AEs the attacker solves the following optimization problem:

$$\max_{\delta} \sum_{d \in D} L_d(x+\delta, y; d) \quad \text{subject to: } ||\delta||_p \leq \epsilon \tag{3}$$

where D is the set of all possible defenses (models) under consideration in the attack, L_d is the loss function associated with defense $d \in D$, δ is the adversarial perturbation, and (x, y) represents the original input with corresponding class label. This formulation of the optimization problem allows the attacker to attack single or multi-model classifiers. The magnitude of this perturbation δ is typically limited by a certain l_p norm. In this paper, we analyze the attacks and defenses using the l_∞ norm. When building our GaME framework, we employ a number of different white-box adversarial attacks. In this section, we briefly discuss the attacks previously developed in the literature (related work). We also give our justification for including them in our analyses.

Momentum Iterative Method [11]: is an attack that can be considered a modified version of PGD [25] that includes a momentum term in the AE update step: $x_{adv}^{(i+1)} = P(x_{adv}^{(i)} + \epsilon_{step} \cdot \text{sign}(g_i))$ where P is the projection operation that ensures that the AE does not exceed the maximum noise bounds, ϵ_{step} is the step size used in the attack and $x_{adv}^{(0)} = x$. The gradient term g_i is computed with respect to the current gradient and previous gradients:

$$g_i = \mu \cdot g_{i-1} + \frac{\nabla_{x_{adv}^{(i)}} L(x_{adv}^{(i)}, y)}{||\nabla_{x_{adv}^{(i)}} L(x_{adv}^{(i)}, y)||_1} \tag{4}$$

In Equation 4 μ is a weighting factor, L corresponds to the loss function used in the attack, y is the correct class label for the original example x and $\nabla_{x_{adv}^{(i)}}$ is the partial derivative of the loss function with respect to the current AE $x_{adv}^{(i)}$.

Why we selected it: It has been shown that MIM has higher levels of transferability over many white-box attacks which do not use momentum, such has PGD [27]. Thus MIM was used in this work for the transferability experiments as a strong baseline for comparison against APGD. We also compare MIM with the new attacks proposed in this work (which we develop later).

Auto-Projected Gradient Descent : [9] The APGD attack can be considered a step size free version of PGD that iteratively updates its starting step size based on changes in the loss function that occur over previous steps. The main AE update equations in APGD are as follows:

$$z^{(i+1)} = P\left(x^{(i)} + \eta^{(i)} \frac{\partial L}{\partial x^{(i)}}\right) \tag{5}$$

$$x^{(i+1)} = P(x^{(i)} + \alpha \cdot (z^{(i+1)} - x^{(i)}) + (1-\alpha) \cdot (x^{(i)} - x^{(i-1)})) \tag{6}$$

where $\alpha \in [0, 1]$ is a hyperparameter that controls the influence of previous updates on the current update, $\eta^{(i-1)}$ is the learning rate at the current iteration and P is the projection operation. The projection operation P is a function used to ensure that the noise added to the adversarial image does not exceed ϵ. For the l_∞ norm, the projection operation has the following form: $P(x, x_{adv}, \epsilon) = \max(\min(x_{adv}, x+\epsilon), x-\epsilon)$. For notational simplicity [9] the projection operation $P(x, x_{adv}, \epsilon)$ is typically written as $P(x_{adv})$.

Why we selected it: APGD has frequently been used a standard benchmark when testing defenses against non-adaptive attacks. In this paper and many others it has shown large attack success rate improvements over other, frequently used, white-box attacks like MIM, PGD, and FGSM.

Other White-Box Attacks: In the literature there are many other examples of white-box attacks that iteratively generate AEs such as PGD [25], the Carlini and Wagner attack [7] and the Elastic-net attack [8] just to name a few. It is important to note that it has been well established in the adversarial machine learning literature [5] that applying many nearly-identical attacks is not useful for evaluation. Hence we focus on APGD, MIM and three new adaptive attacks that we develop specifically for attacking the models and defenses that we analyze in this work.

IV. NEW ADAPTIVE WHITE-BOX ATTACKS

It is crucial for both the attacker and defender to consider the strongest possible adversary when playing the AEs game. Thus, we propose three new adaptive white-box attacks for targeting randomized defenses. The first attack is designed to work on single, randomized defenses and is called the Momentum Iterative Method over Expectation (MIME). To the best of our knowledge, MIME is the first white-box attack to achieve a high attack success rate ($> 70\%$) against the defense TiT. MIME is also capable of achieving a high attack success rate against BaRT, even when non-differentiable transformations are implemented as part of the defense.

Our second attack, is designed to generate AEs that work against multiple type of defenses simultaneously. This compositional attack is called, the Multi-Agent Gradient Expectation attack (MAGE). Lastly, we propose a modified version of MAGE called MAGE-D which is designed to target defenses utilizing consensus voting adversarial detection techniques. In the next section we discuss our new adaptive white-box attacks in detail.

TABLE 1. Performance of the MIME attack against CIFAR-10 randomized defenses TiT and BaRT with $\epsilon_{max} = 0.031$. Lower robustness signifies that the attack is more effective. It can clearly be seen that MIME outperforms both APGD and MIM on these two randomized defenses.

Attack	BaRT-1	BaRT-5	BaRT-10	TIT (BiT/ViT)	TiT (VGG/RN)
MIME-10	**3.18%**	15.5%	43.2%	10.1%	24.9%
MIME-50	4.3%	**8.22%**	**23.2%**	**8.3%**	**23.3%**
MIM	6.7%	39.5%	59.5%	52%	58.9%
APGD	8.9%	47.7%	70.8%	68.2%	40.7%
Clean	98.4%	95.3%	92.5%	90.1%	76.6%

A. MOMENTUM ITERATIVE METHOD OVER EXPECTATION

In this subsection we discuss the motivation and the mathematics for our first novel adaptive white-box attack, the Momentum Iterative Method over Expectation (MIME). Static white-box attacks such as PGD [25] attack often perform poorly against randomized defenses such as BaRT or TiT. In [48] they tested the TiT defense against an attack designed to compensate for randomness, the Expectation over Transformation attack (EOT) attack [3]. However, it was shown that the EOT attack performs poorly against TiT (e.g. 20% or worse attack success rate). For attacking BaRT, in [42] they proposed a new white-box attack to break BaRT. However, this new attack requires that the image transformations used in the BaRT defense to be differentiable. This requirement is a deviation and a property which the original BaRT implementation does not obey. Thus we develop a new adaptive attack to effectively break BaRT, TiT, and other randomized defenses while not requiring additional restrictions on the adversarial model or defense setup.

We develop a new white-box attack specifically designed to work on defenses that inherently rely on randomization, like Barrage of Random Transforms (BaRT) [36] and Trash is Treasure (TiT) [48]. Our new attack is called the Momentum Iterative Method over Expectation (MIME). The attack "mimes" the transformations of the defender in order to more precisely model the gradient of the loss function with respect to the input after the transformations are applied. To this end, MIME utilizes two effective aspects from earlier white-box attacks: momentum from the Momentum Iterative Method (MIM) [11] attack, and repeated sampling [3] from the Expectation Over Transformation (EOT) attack. The AE is iteratively generated in MIME as follows:

$$x_{adv}^{(i+1)} = x_{adv}^{(i)} + \epsilon_{\text{step}} \cdot \text{sign}(g^{(i)}) \tag{7}$$

where the attack is computed iteratively with $x_{adv}^{(0)} = x$. Additionally, $g^{(i)}$ is the momentum based gradient of the loss function with respect to the input at iteration i and is defined as:

$$g^{(i)} := \gamma g^{(i-1)} + \mathbb{E}_{t\sim T}[\frac{\partial L}{\partial t(x_{adv}^{(i)})}] \tag{8}$$

where γ is the momentum decay factor hyperparameter and t is a random transformation function drawn from the defense's transformation distribution T. In practice $g^{(i)}$ is approximated using N Monte Carlo samples per input x:

$$g^{(i)} \approx \gamma g^{(i-1)} + (\frac{1}{N}\sum_{j=0}^{N}\frac{\partial L}{\partial t_j(x_{adv}^{(i)})}) \tag{9}$$

Empirical Results: In Table 1, we show experimental results for the MIME attack on CIFAR-10 randomized defenses (TiT and BaRT). For each defense we use 1000 clean, classwise balance examples that are correctly recognized by the defense. All attacks use a maximum perturbation $\epsilon_{max} = 0.031$ It can clearly be seen that MIME has a higher attack success rate than both APGD [9] and MIM [11]. Our attack is additionally able to reliably break the otherwise robust BaRT and TiT defenses – dropping the robustness of BaRT-10 by 36.3% and TiT (BiT/TiT) by 43.7%.

Algorithm 1 MAGE

Input: clean sample x, step size ϵ_{step}, maximum perturbation ϵ_{max}, coefficent learning rate r, true label y, boolean *targeted*.
Initialize: $G_{blend} = \hat{0}$
Define: $g(m, x) = \sum_{j=0}^{N_{samp}} \frac{\partial L_m}{\partial t_m(x_{adv}^{(i)})}$
For i in range 1 to N_{iter} do:
  $G_{blend} = \gamma \cdot G_{blend} + (\sum_{m=1}^{M} \alpha_m^{(i)} \odot g(m, x_{adv}^{(i)}) \odot \phi_m)$
  If *targeted*
    $x_{adv}^{(i+1)} = x_{adv}^{(i)} - \epsilon_{step} \cdot \text{sign}(G_{blend})$
  else
    $x_{adv}^{(i+1)} = x_{adv}^{(i)} + \epsilon_{step} \cdot \text{sign}(G_{blend})$
  $x_{adv}^{(i+1)} = P(x_{adv}^{(i)}, x, \epsilon_{max})$
  For m in range 1 to M:
    $\frac{\partial x_{adv}^{(i)}}{\partial \alpha_m^{(i)}} \approx u\epsilon_{step}\text{sech}^2(u\sum_{m=1}^{M} g(m, x_{adv}^{(i)}) \odot g(m, x_{adv}^{(i)}))$
    $\frac{\partial F}{\partial \alpha_m^{(i)}} = \frac{\partial F}{\partial \alpha_{adv}^{(i)}} \odot \frac{\partial x_{adv}^{(i)}}{\partial \alpha_m^{(i)}}$
    $\alpha_m^{(i+1)} = \alpha_m^{(i)} - r\frac{\partial F}{\partial \alpha_m^{(i)}}$
Output: $x_{adv}^{(M)}$

B. MULTI-AGENT GRADIENT EXPECTATION ATTACK

The next adaptive attack we develop is called the Multi-Agent Gradient Expectation attack (MAGE) which is specifically adapted to work on defenses using an ensemble of models. It is important to note that single model white-box attacks are not effective when ensembles are employed that contain diverse architctures [27]. Therefore, the use of multi-model attacks are necessary to achieve a high attack success rate when dealing with ensembles that contain both CNN and non-CNN model architectures, like the Vision Transformer (ViT) [12] and Spiking Neural Network (SNN) [13]. This is because AEs generated by single model white-box attacks generally do not transfer well between CNNs, ViTs and SNNs [27], [49]. In addition it is an open question if multi-model attacks can be effective against the current state-of-the-art defenses. In this paper, we expand the idea of a

multi-model attack to include not only different architecture types, but also different defenses. The generalized form of the multi-model attacker threat model is given in Equation 3.

For an input and class label pair (x, y), set of defenses D, and non-detector voting function f, an untargeted multi-model attack is considered successful if:

$$(f(x+\delta, D) \neq y) \wedge (||\delta||_p \leq \epsilon). \tag{10}$$

Equation 10 demonstrates that an untargeted attack on an ensemble based defense system is successful when the voting function f results in a misclassification and the noise used to create the adversarial example remains within the acceptable range ϵ. One formulation of the multi-model attack is the Auto Self-Attention Gradient Attack (Auto-SAGA) [49]. Auto-SAGA iteratively attacks combinations of ViTs, SNNs and CNNs:

$$x_{adv}^{(i+1)} = x_{adv}^{(i)} + \epsilon_{step} \cdot \text{sign}(G_{blend}(x_{adv}^{(i)})) \tag{11}$$

where ϵ_{step} is the step size used in the attack. In the original formulation of Auto-SAGA, G_{blend} was a weighted average of the gradients of each model $d \in D$. By combining gradient estimates from different models, Auto-SAGA is able to create AEs that are simultaneously misclassified by multiple models. One limitation of Auto-SAGA attack is that it does not account for defenses that utilize random transformations. Motivated by this, we can integrate the previously proposed MIME attack into the gradient calculations for Auto-SAGA. We denote this new attack as the Multi-Agent Gradient Expectation attack (MAGE). Both SAGA and MAGE use the same iterative update (Equation 11). However, MAGE uses the following gradient estimator:

$$G_{blend}(x_{adv}^{(i)}) = \gamma G_{blend}(x_{adv}^{(i-1)}) + \sum_{k \in D \setminus R} \alpha_k^{(i)} \phi_k^{(i)} \odot \frac{\partial L_k}{\partial x_{adv}^{(i)}} + \sum_{r \in R} \alpha_r^{(i)} \phi_r^{(i)} \odot (\mathbb{E}_{t \sim T}[\frac{\partial L_r}{\partial t(x_{adv}^{(i)})}]) \tag{12}$$

In Equation 12, the two summations represent the gradient contributions of sets $D \setminus R$ and R, respectively. Here we define R as the set of randomized defenses and D as the set of all the defenses being attacked. In each summation ϕ is the self-attention map [1] which is replaced with a matrix of ones for any defense that does not use attention based models (ViTs). α_k and α_r are the associated weighting factors for the gradients for each deterministic defense k and randomized defense r, respectively. Details of how the weighting factors are derived are given in [49].

C. DETECTION ADAPTED MULTI-AGENT GRADIENT EXPECTATION ATTACK

In order to adapt to detector based ensemble defenses we propose a modification to the MAGE attack. Our new method is called the Detection Adapted Multi-Agent Gradient Expectation attack, or simply MAGE-D. The psuedocode for MAGE-D is given in Algorithm 2. Our new attack exploits the detector's voting scheme by implementing a series of targeted MAGE attacks, thus allowing the attacker to more reliably achieve a plurality, or even majority, of votes for the wrong class label. At a high level, the attack first performs an untargeted MAGE attack and checks if the adversarial example x_{adv} was able to successfully evade detection based on the following criteria:$(f^d(x_{adv}, M) \neq y) \wedge (f^d(x_{adv}, M) \neq \perp)$. The detector based voting function is denoted as f^d, the number of models used in the detection is M and $\perp$ represents the adversarial class label. Under the detection framework, an attack is considered unsuccessful if the final adversarial example is labeled as adversarial: $f^d(x_{adv}) = \perp$.

If the initial MAGE attack on the detection defense is unsuccessful, MAGE-D checks which models were fooled by the untargeted attack (producing label $y' \neq y$). MAGE-D then performs a targeted attack with label y'. The targeted attack will then attempt to push all other models in the ensemble to also classify the AE as y', making it the plurality label.

TABLE 2. Comparison between MAGE and MAGE-D when attacking detector models utilizing vanilla voting models for CIFAR-10. All attacks use $\epsilon_{max} = 0.031$. The MAGE and MAGE-D columns represent the robust accuracy of the detector defense when evaluating samples generated by the MAGE and MAGE-D defenses respectively.

Defense Type	Max APGD	Adaptive APGD	MAGE	MAGE-D
With Detection	18.4%	18.8%	29.4%	**43.2%**
No Detection	41.2%	45.2%	46.6%	**59.6%**

1) EMPIRICAL RESULTS

We evaluated the MAGE and MAGE-D attacks against an ensemble defense using vanilla (undefended) ViT and Resnet classifiers. The attacks were compared against multi-model versions of APGD we refer to as Max APGD and Adaptive APGD [49]. Both attacks perform the single model version of APGD against each model in the defense. Max APGD then chooses every sample from the attack with the highest attack success rate against the ensemble. Adaptive APGD instead chooses adversarial examples on a per-sample basis, resulting in an adaptive mixture of samples from each attack. In Table 2 we numerically present our comparison. Each attack uses 1000 clean classwise balanced samples that are correctly recognized and a maximum perturbtation of $\epsilon_{max} = 0.031$. Interestingly, we find that neither max or adaptive APGD are able to achieve high levels of attack success against the detector based defense. MAGE-D, on the other hand, is able to improve attack success rate by 14.4% against the non-detector and by 24.4% against the detector based ensemble. MAGE also sees improvement over the APGD based attacks, with 1.4% and 10.6% improvements respectively.

V. DEFENSE TRANSFERABILITY ANALYSIS

Adversarial transferability refers to the phenomena in which AEs generated to attack one model are also misclassified by a different model. Adversarial transferability studies have

Algorithm 2 MAGE-D

Input: clean sample x, true label y, models M.
Let $\sigma(m, x)$ be a function that gets the output logits of a classifier $m \in M$ given input x
Initialize: y', x'_{adv}, and s to be lists indexed by $m \in M$
$x_{adv} = \text{MAGE}(x, y, \text{False})$ ▷ Try unlabeled attack
If $f^d(x_{adv}, M) \neq y$ and $f^d(x_{adv}, M) \neq \perp$
  return x_{adv} ▷ Stop if attack was successful
For $m \in M$
  $y'_m = m(x_{adv})$
  If $y' \neq y$
    $x'_{adv,m} = \text{MAGE}(x, y'_m, \text{True})$ ▷ Perform targeted MAGE with label y'_m
    $s_m = \sigma(f^d(\cdot, M), x'_{adv,m})$
$m^* = \arg\max_{m \in M}[\max_{y'} s_{m,y' \neq y}]$ ▷ Get index of most successful sample
return x'_{adv,m^*}

been done on a variety of machine learning models [24], [27], [49]. However, to the best of our knowledge, adversarial transferability between different state-of-the-art defenses has not been conducted. This transferability property is of significant interest because a lack of transferability between different defenses may indicate a new way to improve adversarial robustness.

In both Tables 3 and 4 we use the following abbreviation for each defense: B refers to the Barrage of Random Transforms defenses [36] and the number next to B corresponds to the number of transformations used, e.g. B5 refers to Barrage of Random Transforms implemented with 5 randomized image transformations used. RF corresponds to a ResNet model trained with Friendly Adversarial Training (FAT) [52]. VF corresponds to a ViT model with Friendly Adversarial Training. D refers to the diffusion model based adversarial training [47]. ST [37] denotes the Spiking Neural Network trained using the weight transfer approach and SB [13] denotes the Spiking Element Wise (SEW) ResNet. Lastly, BVT and VRT refer to the Trash is Treasure defense(TiT) [48] which is a defense built with two models. TiT built with a BiT [19] and ViT model is abbreviated as BVT and VRT is TiT built with a VGG [41] and ResNet [17] model respectively.

In our study we first chose 1000 classwise balanced, clean images from the testing set of CIFAR-10. We additionally constrained these 1000 samples to be those which are correctly identified by every model in Table 4. For random transform defenses, each image is classified correctly with a probability of at least 98%. We then attacked each defense with APGD, every randomized defense with MIME, and every non-randomized defense with MIM. In Table 3, we show the different defenses we analyze in this paper and the best attack on each of them from the set of attacks MIM [11], APGD [9] and MIME (proposed in this work).

TABLE 3. (CIFAR-10) Single defense implementations with the corresponding strongest attack on the defense and the clean accuracy of the defense. The robust accuracy is measured using 1000 adversarial examples. The examples are classwise balanced and also correctly recognized by all the defenses in their original clean form.

Defense	Best Attack	Clean Acc	Robust Acc
B1	MIME	98.4%	3.4%
B5	MIME	95.3%	15.0%
B10	MIME	92.5%	43.5%
RF	APGD	81.9%	52.0%
VF	APGD	92.4%	25.0%
D	APGD	96.3%	70.0%
ST	APGD	91.5%	0.0%
SB	APGD	81.2%	1.6%
BVT	MIME	90.1%	8.6%
VRT	MIME	76.6%	26.2%

A. ANALYSIS OF TRANSFERABILITY

From each defense we chose the adversarial samples generated by the attack with the highest attack success rate to be used in the transfer study shown in Table 4. We then have the other models (which were not used in the attack) evaluate these adversarial examples. From the transferability Table 4 it becomes clear that there is generally very low transferability (high robustness) between certain pairs of defenses. For example, if we take the first row of the table, we can see how accurately other defenses evaluate the adversarial examples generated by BaRT-1 (B1). In the first row, we can see clear examples of low transferability, where both the ViT trained FAT model (VF) and the ResNet trained FAT model (RF) can correctly identify B1 adversarial examples more than 99% of the time. In contrast to this, defenses which share architectures, unsurprisingly, often have relatively high levels of transferability. For instance for Spiking Neural Networks this trend occurs for ST and SB where, ST only recognizes adversarial examples generated by SB 64.9% of the time. For the BaRT defenses, this trend is even more pronounced. The B1 defense only recognizes adversarial examples generated by the B10 defense 18.6% of the time.

These results in turn motivate the development of a game theoretic framework for both the attacker and defender. For the attacker, this prompts the need to use multi-model attack like MAGE as there are cases where no single attack (APGD, MIM or MIME) is ubiquitous. For the defender, these results highlight the opportunity to increase robustness by taking advantage of the low levels of transferability between defenses through the implementation of a randomized ensemble defense.

VI. GAME THEORETIC MIXED EXPERTS (GaME)

In this section we derive our framework, Game theoretic Mixed Experts (GaME), for approximating a Nash equilibrium in the adversarial examples game. In comparison to other works [2], [4], [22], [29], [31], [34] we take a more discretized approach and solve the approximate version of the adversarial examples game. This ultimately leads to the

TABLE 4. (CIFAR-10) Transferability study between all defenses in this paper. Here each value represents the robust accuracy of the column defense when evaluating samples generated by attacking the row model (defense). Thus, smaller values represent a higher level of transferability since more samples are incorrectly classified by the evaluating model (defense).

Transferability Between Defenses (CIFAR-10)										
Evaluation Model → Attacked Model ↓	B1	B5	B10	D	RF	VF	ST	SB	BVT	VRT
B1	-	45.8%	67.2%	95.3%	99.6%	99.2%	84.1%	93.1%	95.9%	89.9%
B5	4.3%	-	44.4%	94.6%	98.7%	97.9%	71.3%	90.5%	89.7%	89.0%
B10	18.6%	4.0%	-	92.9%	83.2%	90.8%	66.5%	77.5%	82.5%	70.2%
D	74.2%	72.4%	69.3%	-	66.2%	71.7%	58.6%	64.2%	71.1%	78.2%
RF	91.8%	88.4%	82.2%	88.1%	-	87.9%	68.6%	69.8%	80.2%	66.6%
VF	52.4%	55.7%	55.7%	89.9%	86.0%	-	63.0%	62.9%	21.7%	74.0%
ST	91.5%	90.5%	89.4%	94.5%	98.9%	98.9%	-	91.8%	91.8%	91.3%
SB	92.2%	89.1%	86.3%	94.2%	93.9%	95.2%	64.9%	-	82.9%	84.7%
BVT	60.4%	69.9%	75.1%	95.1%	98.2%	95.6%	78.1%	90.0%	-	90.8%
VRT	86.8%	83.4%	89.8%	89%	82.3%	87.2%	76.0%	71.0%	78.8%	-

creation of a finite, tabular, zero-sum game that can be solved in polynomial time using linear programming techniques.

A. BACKGROUND-MULTI-MODEL VOTING SCHEMES

In the GaME framework we will study the effectiveness and optimization of multi-model voting schemes. Formally we let the defender p_D have access to a set of defenses $D \subset \Theta$ which it can use to classify a given input $x \in \mathcal{X}$ with true class label $y \in \mathcal{Y}$. Here Θ is the set of all possible model parameters, $\mathcal{X}$ is the latent input space, and $\mathcal{Y}$ is the space of output labels. The defender then computes their output as follows:

$$\hat{y} = f(x, U) \tag{13}$$

where $\hat{y}$ is the predicted class label, f is the defender's chosen multi-model voting function, and $U \subseteq D$ is the defender's chosen subset of defenses to use in classification. In this paper we first focus on two, non-detector voting functions: the plurality vote (f^h), and the softmax probability vote (f^s) [42]:

$$f^h(x, U) = \arg\max_{y \in \mathcal{Y}} \sum_{d \in U} \mathbb{1}\{y = \arg\max_{j \in \mathcal{Y}} d_j(x)\} \tag{14}$$

$$f^s(x, U) = \arg\max_{y \in \mathcal{Y}} \frac{1}{|U|} \sum_{d \in U} \sigma(d(x)) \tag{15}$$

where $\mathbb{1}$ is the indicator function, σ is the softmax function, $d(x)$ is the logit vector of defense d evaluated on input x, and $d_j(x)$ represents the j^{th} logit output. In practice we find both of these voting schemes to perform similarly, though softmax voting can become biased towards high confidence predictions from individual models, resulting in lower robustness. We additionally consider detection based multi-model defenses in which the defender p_d is allowed to output the label $\perp$ which marks input x as adversarial. This allows the defender to either filter out or mitigate the impact of adversarial examples at the cost of some accuracy on clean samples. In this work we utilize a plurality voting scheme in which inconclusive votes are given the $\perp$ label. In particular, we define detection based voting f^d as:

$$f^d(x, U) = \begin{cases} y'_i & \text{if } \exists\, V_{y'_i} : \forall y'_j \neq y'_i,\ V_{y'_i} > V_{y'_j} \\ \perp & \text{otherwise} \end{cases} \tag{16}$$

where $V_{y'_i}$ represents the total number of votes for class label y'_i. Under this defense condition a successful attack consequently becomes more restrictive, requiring the attacker to both induce an incorrect class label and evade the $\perp$ label:

$$(f^d(x_{adv}, U) \neq y) \wedge (f^d(x_{adv}, U) \neq \perp) \tag{17}$$

where (x, y) is the clean input and correct class label used to generate adversarial example x_{adv}. This means that the adversary must not only induce incorrect labels amongst all defenses $d \in U$, but must also induce the same class label between a plurality of models. This requires a special attack formulation, like MAGE-D, which we discuss in Section IV.

B. THE ADVERSARIAL EXAMPLES GaME

We build upon and discretize the adversarial examples game explored in [29]. The adversarial examples game is a zero-sum game played between two players: the attacker, p_A, and the defender p_D. Let $\mathcal{X}$ be the input space and $\mathcal{Y}$ the output space of p_D's classifiers, and let Θ represent the space of classifier parameters. Additionally, let $P_{\epsilon,\mathcal{X}} = \{x \in \mathcal{X} : ||x||_p \leq \epsilon\}$ be the set of valid adversarial perturbations for norm p and $\epsilon \in \mathbb{R}^+$. Let $A^*_\epsilon = \{(f : \Theta \times \mathcal{X} \times \mathcal{Y} \to P_{\epsilon,\mathcal{X}})\}$ be the set of all valid attack functions. The goal of p_A is to choose $a \in A^*_\epsilon$, which maximizes the expected loss of p_D's classifier, θ, given some pair of input and ground truth label $(x, y) \in \mathcal{X} \times \mathcal{Y}$. The goal of p_D is to minimize this loss through its choice of $\theta \in \Theta$. We can thus formulate the adversarial examples game as a mini-max optimization problem:

$$\inf_{\theta \in \Theta} \sup_{a \in A^*_\epsilon} \mathbb{E}_{(x,y)\sim \mathcal{X}\times\mathcal{Y}}[L(x + a(\theta, x, y), y; \theta)] \tag{18}$$

Due to the vastness of Θ and A^*_ϵ, solving this optimization problem directly is currently computationally intractable. To this end, in the next subsections we will formulate GaME$_n$ and Ada-GaME$_n$ which discretize Θ and A^*_ϵ by enlisting a set of state-of-the-art attacks and defenses.

C. THE GENERALIZED GaME FRAMEWORK

The goal of the GaME framework is to find an approximate solution to the adversarial examples game through the

implementation of a set of attacks and defenses which will serve as experts for p_A and p_D.

Let $A' \subset A^*_\epsilon$ be a subset of all valid adversarial attack functions chosen by p_A. Additionally, let $D' \subset \Theta$ be a set of defense classifiers chosen by p_D. We further impose that all $a \in A'$ are white-box attacks (see Section III for our adversarial threat model) and that A', D' are finite.

It is important to note that each $a \in A'$ is a function of some classifier, $\theta \in \Theta$, in addition to the input and ground truth label. Due to this it is possible for p_A to chose to attack defense $d \in D'$ with attack $a \in A'$, while p_D chooses to evaluate the sample using defense $d' \in D'$ where $d \neq d'$. Therefore, for convenience, we will define a new, more general set of attack strategies for p_A:

$$A \subseteq \{(g : \mathcal{X} \times \mathcal{Y} \to P_{\epsilon,\mathcal{X}}) : \\ g(x, y) = a_i(U, x, y),\ a_i \in A',\ U \subseteq D\} \tag{19}$$

where we extend the definition of $A' \subseteq A^*_\epsilon$ to attack functions, such as MAGE, that can take subset of defense parameters $U \subseteq D$ as input (see Equation 3 for our multi-model attack formulation).

In addition to extending the capabilities of the attacker via multi-model attacks, we will further extend the defender's options by allowing the usage of multi-model voting schemes. In the general GaME framework, the defender can choose a parameter, n, which determines their maximum ensemble voting size.

For $n > 1$, p_D can calculate their final prediction based upon the output logits of multiple $d \in D'$ evaluated on the same input x. For this to occur, p_D must also choose a function to map the output of multiple defenses to a final prediction. Formally, the strategy set of p_D becomes $D = P_n(D') \times F$, where F is a set of prediction functions and $P_n(D')$ is defined as follows.

$$P_n(D') = \{U : U \subseteq D,\ |U| \leq n\} \tag{20}$$

Thus we will let D be the strategy set of p_D, and A be the strategy set of p_A. These defenses are used in multi-model prediction functions and targeted by attacks explored in Section IV.

D. GaME-N

Here we present the GaME framework, GaME$_n$, which aims to find an approximate solution to the adversarial examples game when the defender and attacker can each implement an ensemble of defenses and attacks respectively.

First, rather than optimizing over the defender's loss, we instead proceed by optimizing over the defender's robust accuracy. In this way the defender's goal is to maximize their robust accuracy, while the attacker aims to minimize it, i.e. maximize their attack success rate. Thus, for notational convenience, we will define the following payoff matrix, R^D, for the defender:

$$R^D_{U_j,a_k} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{P}(f(x_i + a_k(x_i, y_i), U_j) = y_i) \tag{21}$$

where $U_j \in P_n(D')$, f is the voting function, and the attack strategy is $a_k \in A$. Note that instead of calculating R^D over the whole of $\mathcal{X} \times \mathcal{Y}$, which is currently computationally intractible for sufficiently large datasets, we instead approximate its values by taking N Monte-Carlo samples with respect to $(x_j, y_j) \in \mathcal{X} \times \mathcal{Y}$. Thus $R^D_{U_j,a_k}$ represents the expected robust accuracy, i.e., the payoff, for p_D if they play strategy U_j and p_A plays strategy a_k. The payoff matrix for p_A, R^A, is defined as $1 - R^D$.

As previously explored in [2], [29], limiting ourselves to pure, deterministic strategies severely inhibits the strength of both the attacker and defender. Thus we create mixed strategy vectors, $\lambda^A \in \mathbb{R}^{|A|}$ and $\lambda^D \in \mathbb{R}^{|D|}$, for p_A, p_D defined by:

$$\mathbb{P}(\{a_i \in A : a = a_i\}) = \lambda^A_i : ||\lambda^A||_1 = 1 \\ \mathbb{P}(\{d_i \in D : d = d_i\}) = \lambda^D_i : ||\lambda^D||_1 = 1 \tag{22}$$

where $a \in A$ and $d \in D$ are random variables. With these mixed strategy vectors we can then reformulate the adversarial examples game as a mini-max optimization problem over p_D's choice of λ^D and p_A's choice of λ^A:

$$\max_{\lambda^D} \min_{\lambda^A} [\langle \lambda^D,\ R^D,\ \lambda^A \rangle] \tag{23}$$

where $\langle \cdot, \cdot, \cdot, \rangle$ represents the inner product. From this we can create a finite, tabular, zero-sum game defined by the following game-frame in strategic form:

$$\langle \{p_A, p_D\},\ (A, D),\ O,\ r \rangle \tag{24}$$

where $O = \{R^D_{d,a}\ \forall a \in A,\ d \in D\}$ and r is a function $r : A \times D \to O$ defined by $f(d, a) = R^D_{d,a}$. Because this is a finite, tabular, zero-sum game, it has a Nash-Equilibrium [30]. Let R^D be the payoff matrix for p_D. It then becomes the goal of p_D to maximize their guaranteed, expected payoff. Formally, p_D must solve the following optimization problem:

$$cc \max_{r^*;\lambda^D} r^* \text{ subject to } \lambda^D R^D \geq (r^*, \cdots, r^*) \text{ and } ||\lambda^D||_1 \leq 1 \tag{25}$$

This optimization problem is a linear program, the explicit form of which we provide in the appendix. All linear programs have a dual problem, in this case the dual problem finds a mixed Nash strategy for p_A. This can be done by replacing $\lambda^D R^D$ with $\lambda^A (R^A)^T$. In the interest of space we give the explicit form of the dual problem in the appendix as well. These linear programs can be solved using polynomial time algorithms.

E. Ada-GaME

Adaptive attacks such as AutoAttack [9] have been frequently used as a benchmark for defense robustness. The main effectiveness of adaptive attacks comes from their ability to create multiple adversarial examples per clean sample and choose one which has the highest attack success rate. In this way, the attacker is able to exploit the combined power of multiple attacks rather than choosing the best single attack.

Given a defense classifier, d, a set of attack functions A, and a single clean sample with true label (x_i, y_i), an attacker is able to create a set of adversarial samples, X_i. From this set the attacker can choose a single adversarial example for the attack.

$$X_i = \{x_k^i \in \mathcal{X} : x_k^i = x_i + a_k(x_i, d),\ a_k \in A\} \tag{26}$$

For notational convenience we will define an accuracy matrix, R^i, defined as follows for arbitrary $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$:

$$R^i \in [0,1]^{|A|\times|D|} : R^i_{k,j} = \mathbb{P}(d_j(x_k^i) = y_i) \tag{27}$$

where $d_j \in D$, the set of all defense classifiers the defender has access to. Within the adaptive attack adversarial threat model, we assume a strong adversary which knows the defenses D, along with the defender's mixed strategy vector, λ^D, as defined in Equation 22. With this threat model we can formulate the adversarial examples game minimax problem as follows:

$$\min_{\lambda^D} \mathbb{E}_{(x_i,y_i)\sim\mathcal{X}\times\mathcal{Y}}[\max_k[1 - R^i \cdot \lambda^D]_k] : ||\lambda^D||_1 = 1 \tag{28}$$

we use the notation $\max_k[v]_k$ to denote finding the index k of vector v which results in the maximum value. Note that by optimizing over $1 - R^i \cdot \lambda^D$ the attacker maximizes their attack success rate while the defender minimizes it. We can approximate this optimization problem by taking N Monte-Carlo samples with respect to $(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$:

$$\min_{\lambda^D} \frac{1}{N}\sum_{i=1}^{N} \max_k[1 - R^i \cdot \lambda^D]_k : ||\lambda^D||_1 = 1 \tag{29}$$

This formulation of the optimization problem for the defender results in a convex, continuous optimization function.

1) PROOF OF Ada-GaME CONVEXITY

We provide a proof for the convexity of Ada-GaME's optimization problem. In particular we will show the following is convex for arbitrary matrices R^i:

$$f(\lambda) = \frac{1}{N}\sum_{i=1}^{N} \max_k[1 - R^i\lambda]_k \tag{30}$$

To prove f is convex one must show:

$$\alpha f(\lambda) + (1-\alpha) f(\lambda') \geq f(\alpha\lambda + (1-\alpha)\lambda') \tag{31}$$

for arbitrary $\lambda, \lambda' \in \mathbb{R}^{|D|}$ and scalar $\alpha \in [0, 1]$. For notational convenience we will also let $\beta = 1 - \alpha$ Thus we can proceed as follows:

$$\begin{aligned}
&\alpha\frac{1}{N}\sum_{i=1}^{N}(\max_{k_1}[1 - R^i\lambda]_{k_1}) + \beta\frac{1}{N}\sum_{i=1}^{N}(\max_{k_2}[1 - R^i\lambda']_{k_2}) \\
&= \frac{1}{N}\sum_{i=1}^{N}\left(\max_{k_1}[\alpha(1 - R^i\lambda)]_{k_1} + \max_{k_2}[\beta(1 - R^i\lambda')]_{k_2}\right) \\
&\geq \frac{1}{N}\sum_{i=1}^{N}\max_k([\alpha(1 - R^i\lambda)]_k + [\beta(1 - R^i\lambda')]_k) \\
&= \frac{1}{N}\sum_{i=1}^{N}\max_k[1 - (\alpha R^i\lambda + \beta R^i\lambda')]_k \\
&= \frac{1}{N}\sum_{i=1}^{N}\max_k[1 - (R^i \cdot (\alpha\lambda + \beta\lambda'))]_k \\
&= f(\alpha\lambda + (1-\alpha)\lambda') \\
&\rightarrow \alpha f(\lambda) + (1-\alpha) f(\lambda') \geq f(\alpha\lambda + (1-\alpha)\lambda') \\
&\quad Q.E.D.
\end{aligned}$$

Therefore, we have proven the convexity of Ada-GaME's optimization problem.

The Ada-GaME optimization problem is also differentiable w.r.t. λ^D at all points with unique maximum values. At all non-differentiable points we can set the derivative to 0. Thus the optimization problem can be solved with multiple different constrained, gradient-based optimization techniques. For this work we chose to implement a form of Projected Gradient Descent (PGD) as λ^D is now constrained to the following manifold in $\mathbb{R}^n$:

$$P^n \subset \mathbb{R}^n : P^n = \{\lambda \in \mathbb{R}^n : ||\lambda||_1 = 1\} \tag{32}$$

Thus, we can define a reduction, or projection from $\mathbb{R}^n\backslash\{0\}$ to P^n as follows:

$$F : \mathbb{R}^n\backslash\{0\} \rightarrow P^n \text{ where } F(x) = \frac{x}{||x||_1} \tag{33}$$

The reduction, F, allows the iterative optimization process to stay within our constraints on λ^D. We can then define the update equation of λ_s^D as:

$$\lambda_{s+1}^D = F(\lambda_s^D - \alpha_s(\nabla_{\lambda^D} L(R, y; \lambda_s^D))) \tag{34}$$

where α is a step size parameter. In our implementation we optimize over the α parameter at each iteration via a binary search: choosing the alpha which, once applied to λ_{s+1}^D, minimizes the attacker's success rate. Optimization terminates once no α value can be found which further decreases the attack success rate. Experimentally, our algorithm can be shown to converge after only a few iterations (< 10).

VII. EXPERIMENTAL RESULTS

Experimentally we analyze our GaME framework on two datasets, CIFAR-10 and Tiny-ImageNet. Our experiments demonstrate the superiority of our proposed framework in terms of W-Robustness. We also provide code related to our experiments on Github here.

We organize the discussion of our experimental results as follows: first we describe the details of the datasets, defenses and attacks. Second, in subsection VII-A we analyze the robustness of single defenses and a uniform distribution ensemble defense. The best single defense and uniform ensemble serve as baseline measurements for comparison to our GaME framework. Third, in subsection VII-B we analyze the results of the GaME famework and compare it to the baseline results. Lastly, in subsection VII-C we analyze the uniform ensemble baseline and the GaME ensemble under a stronger adaptive adversary that runs attacks on a per sample basis.

Datasets: CIFAR-10 is a dataset made up of 50, 000 training images and 10, 000 testing images [20]. Each image is $32 \times 32 \times 3$ (a 32×32 color image) and is one of ten classes. The 10 classes in CIFAR-10 are airplane, car, bird, cat, deer, dog, frog, horse, ship and truck. Tiny-ImageNet has 100, 000 training images and 10, 000 validation images with 200 classes [23]. Each image in Tiny-ImageNet is $64 \times 64 \times 3$. We selected CIFAR-10 because it is a standard dataset widely used in the adversarial machine learning literature as a benchmark for measuring robustness [26], [44], [47]. We selected Tiny-ImageNet due to the challenging nature of the dataset, as well as its widespread use in adversarial machine learning [10], [42].

Defenses: For generating our GaME ensemble results, we use the following set of defenses. For CIFAR-10 we implemented GaME utilizing: BaRT-1 (B1), BaRT-5 (B5), ResNet-164-FAT (RF), ViT-L-16-FAT (VF), diffusion based Wide ResNet (D), SNN Transfer (ST), Backprop SNN (SB), and TiT using ViT and BiT (BVT). For Tiny-ImageNet we implemented GaME utilizing: BaRT-1, BaRT-5, ViT-L-16-FAT, diffusion based Wide ResNet, and TiT using ViT and BiT. We chose the Bit+ViT version of TiT since the original, VGG-ResNet, architecture that was proposed has a significantly lower clean accuracy [48]. BaRT-1 and BaRT-5 were chosen in favor of BaRT-10 due to the computational cost of computing all the necessary attacks on BaRT-10.

Attacks: When running GaME we generate adversarial examples for each individual defense based on which attack is most effective: we use APGD for the non-randomized defenses (ResNet-164-FAT, ViT-L-16-FAT, the diffusion based Wide ResNet, SNN Transfer, and Backprop SNN). We use MIME for the randomized defenses (BaRT-1, BaRT-5 and TiT). We also attack each pair of defenses with MAGE and MAGE-D. This is a total of 64 attacks on CIFAR-10 and 25 attacks on Tiny-ImageNet. Every attack was run with respect to the l_∞ norm. The hyperparameters for each attack are given in Table 5. For each attack we first chose a class-wise balanced set of 1000 clean images from the testing set, of each respective data set. From this subset, we generated 1000 adversarial examples for each attack. We used 800 of these samples to create the payoff matrix R, then evaluated the ensemble using the remaining 200, class-wise balanced samples from each attack.

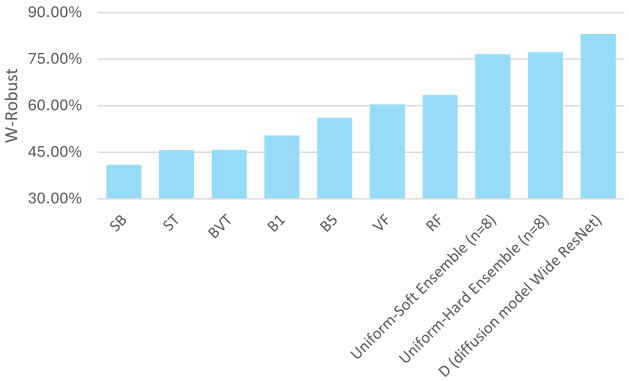

FIGURE 1. (CIFAR-10) W-Robust for baseline defenses (shown in blue). The best baseline is the diffusion model Wide ResNet [47] with a W-Robust value of 83.15%. Full numerical results are given in Tables 6 and 8.

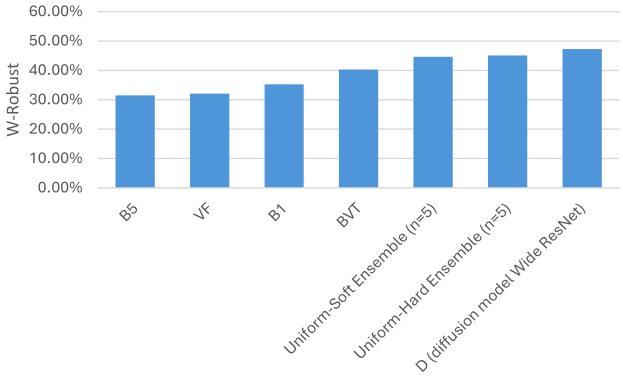

FIGURE 2. (Tiny-ImageNet) W-Robust for baseline defenses (shown in blue). The best baseline is the diffusion model Wide ResNet [47] with a W-Robust value of 47.25%. Full numerical results are given in Tables 7 and 9.

Metrics: For adversarial machine learning defenses we are primarily concerned with two measurements, the clean accuracy and the robustness to adversarial attacks. It may be challenging to compare two defenses when neither of the defenses is optimal in both metrics. E.g., defense A has higher clean accuracy, but defense B has higher robustness. One metric which combines both measurements is the weighted robustness (W-Robust) which averages both clean accuracy and robustness together [16] and is commonly used [10], [53].

In our experiments we give three numbers. First, we report the robustness. This measures the percentage of the 200 adversarial examples created from the validation set, that the defense recognizes correctly. Second, we report the clean accuracy. This is the percentage of all validation samples (unperturbed) that the defense recognizes correctly. Lastly, we report the W-Robust measurement for each defense configuration.

A. BASELINE EXPERIMENTS

The first set of experiments we run are used to establish a baseline for which we can compare the ensemble defense

TABLE 5. Attack parameters. Here γ represents the momentum decay rate, ϵ represents the maximum allowed perturbation magnitude, α represents the weights used in the MAGE algorithm, and N represents the number of EOT samples taken. Note for APGD that the ϵ_{step} value presented is only the initial value and is subject to change according to the attack's algorithm.

Attack	ϵ	ϵ_{step}	Attack Steps	N	γ	Fitting Factor	α Learning Rate
APGD	.031	.005	20	-	-	-	-
MIM	.031	.0031	10	-	.5	-	-
MIME	.031	.0031	10	10	.5	-	-
MAGE	.031	.005	40	4	.5	50	10000
MAGE-D	.031	.005	40	4	.5	50	10000

generated by the GaME framework. In general there are two baselines for which we can compare the GaME frameworks. For our first baseline, we simply determine the single best defense from all the possible defenses we analyzed in this work. For this experiment we run the best individual attack on each single defense and measure the corresponding clean accuracy, robustness and W-Robust. For our second baseline we consider an ensemble defense, in which no probability vector is calculated, i.e., a uniform distribution is used when selecting between each of the defenses.

1) SINGLE DEFENSE BASELINE EXPERIMENTS

In Table 6 and 7 we attack each single defense and report the clean accuracy, robustness and W-Robust metric. Visually the results for these experiments are shown in Figure 1 for CIFAR-10 and Figure 2 for Tiny-ImageNet. It can clearly be seen that in terms of robustness the diffusion based Wide ResNet (D) has the highest value (70%) and the highest W-Robust value (83.5%) for CIFAR-10. Likewise for the Tiny-ImageNet dataset, the diffusion based Wide ResNet (D) is the strongest single defense with a W-Robust value of 47.25%.

2) UNIFORM ENSEMBLE DEFENSE BASELINE EXPERIMENTS

In Table 8 and Table 9 we give results for a uniform ensemble defense. In these tables, n corresponds to the number of defenses used. For example, for Tiny-ImageNet with a defense $n = 3$, this corresponds to randomly selecting 3 out of the 5 defenses every time an input is given to the ensemble. When n of the defenses are selected, voting is performed in two different ways. In Uniform-Soft, the voting is done by adding the softmax outputs of the models together. In Uniform-Hard voting is done by only using the class label that each model produces. From these experiments, it can clearly be seen that for both CIFAR-10 and Tiny-ImageNet, the hard label voting produces the highest W-Robust. For CIFAR-10, the Uniform-Hard with $n = 8$ produces a W-Robust value of 77.25%. For Tiny-ImageNet the Uniform-Hard with $n = 5$ produces the highest W-Robust with a value of 45.10%.

3) BASELINE EXPERIMENTS ANALYSES

From our baseline experiments we can derive several important points. First, the diffusion based Wide ResNet (D) is the best single model defense to compare with the our GaME framework as it achieves the highest W-Robust value

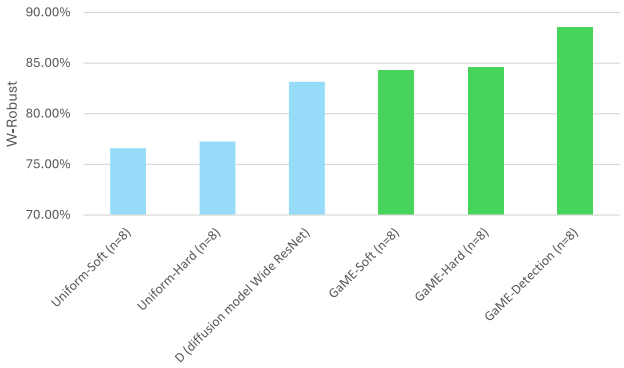

FIGURE 3. (CIFAR-10) W-Robust for baseline defenses (shown in blue) and GaME frameworks (shown in green). The best baseline is the diffusion model Wide ResNet [47] with a W-Robust value of 83.15%. The best GaME ensemble defense is Game-Detection with a W-Robust value of 88.55%.

for both datasets that we tested on. Second, increasing the value of n in uniform ensemble defenses increases the value of W-Robust. This general is intuitive, as the more defenses used in the ensemble, the more difficult it is for the attacker to generate adversarial examples. However, this trend only holds for Uniform-Hard and does not hold for Uniform-Soft when considering robustness. Robustness actually drops for the Uniform-Soft ensemble from 26% when $n = 1$ to 18% when $n = 5$ for the Tiny-ImageNet dataset. This interesting finding bring us to our next point. Third, a Uniform-Hard ensemble is the best performing uniform baseline. This may be due to the fact that in Uniform-Soft the attacker may use a summation of confidences from other models to yield misclassifications. This would explain why for the Tiny-Imagenet ensemble the robustness drops as we increase n. Adding additional vulnerable defenses that contribute soft label confidences actually gives the attacker a higher probability of succeeding. In Uniform-Hard, only binary outputs are used (the output of each model is 1 for the most confident class and 0 for all other classes in the voting scheme). Overall our experiments in this subsection reveal the strongest possible baselines that we can use, to compare to our GaME ensemble framework in the next subsection.

B. GaME EXPERIMENTS

We experiment with the GaME framework for various values of n under a soft label (GaME-Soft), hard label (GaME-Hard) and voting scheme. Numerical results for the standard

TABLE 6. (CIFAR-10) Baseline performance of each single defense.

Defense	B1	B5	RF	VF	ST	SB	BVT	D
Clean	98.40%	95.30%	81.90%	92.40%	91.50%	81.20%	90.10%	96.30%
Robustness	2.50%	17.00%	45.00%	28.50%	0.00%	0.80%	1.50%	70.00%
W-Robust	50.45%	56.15%	63.45%	60.45%	45.75%	41.00%	45.80%	**83.15%**
Best Attack	MIME	MIME	APGD	APGD	APGD	APGD	MIME	APGD

TABLE 7. (Tiny-ImageNet) Baseline performance of each single defense.

Defense	B1	B5	VF	BVT	D
Clean	65.50%	54.00%	60.20%	78.50%	65.50%
Robustness	5.00%	9.00%	4.00%	2.00%	29.00%
W-Robust	35.25%	31.50%	32.10%	40.25%	**47.25%**
Best Attack	MIME	MIME	APGD	APGD	APGD

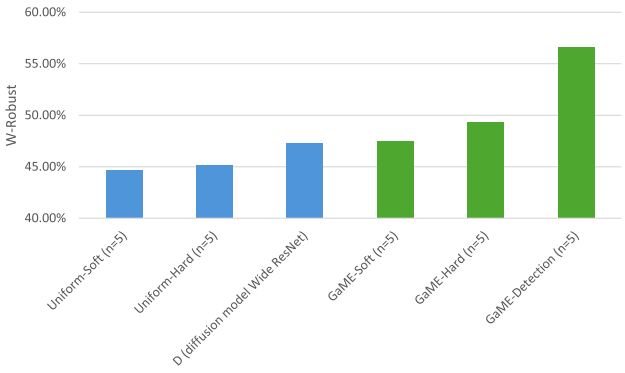

FIGURE 4. (Tiny-ImageNet) W-Robust for baseline defenses (shown in blue) and GaME frameworks (shown in green). The best baseline is the diffusion model Wide ResNet [47] with a W-Robust value of 47.25%. The best GaME ensemble defense is Game-Detection with a W-Robust value of 56.60%.

GaME framework are given in Table 10 for CIFAR-10 and in Table 11 for Tiny-ImageNet. Overall, comparing GaME-Soft and GaME-Hard we can see that the hard label voting performs marginally better than soft label voting across both datasets. For example, for Tiny-ImageNet the highest W-Robust for soft label voting is 47.45% when $n = 5$. For the same n value the W-Robust for hard label voting in GaME is 49.30%. This trend is consistent with what was observed in the baseline experiments where hard label voting for uniform defenses also outperformed soft label voting.

1) GaME-DETECTION

An alternative to standard hard and soft label voting is the GaME framework with detection. As previously detailed in Section VI-A, in the detection scheme n defenses are chosen and hard label voting is done. However, if a plurality of models do not agree on the class label, the defense outputs the adversarial label. The numerical results for GaME-Detection are given in Tables 12. When comparing the three GaME frameworks we can clearly see that Game-Detection has the highest W-Robust for both CIFAR-10 and Tiny-ImageNet as compared to GaME-Soft and GaME-Hard.

2) GaME ENSEMBLES VERSUS BASELINES ANALYSIS

In Figure 3 we show the top three baseline defenses for CIFAR-10, alongside the top three different GaME generated ensembles. Likewise for Tiny-ImageNet, the comparison between baselines and GaME ensembles is shown in Figure 4. The results show that the GaME framework improves upon the robustness of the baseline uniform probability distribution ensemble while maintaining a high level of clean accuracy. In particular, in Figure 3 we see that all versions of GaME are able to out-perform the uniform defense strategy. This emphasizes the importance of not only the voting function, but also the weighting of each model within the ensemble. Specifically, with adversarial detection, GaME is able to outperform the Uniform-Hard by 30.1% in terms of robustness for CIFAR-10 while only dropping clean accuracy by 7.5%. When compared to the best single model defense baseline, GaME-Detection has 7.2% less clean accuracy, but is 18% more robust for CIFAR-10.

For conciseness, the W-Robust measurement can be used as a single metric to compare the defenses. When analyzing W-Robust, it can clearly be seen that GaME is superior to the other defenses for both datasets. For CIFAR-10, GaME-Detection outperforms the strongest baseline (diffusion model Wide ResNet) by 5.4%. For Tiny-ImageNet GaME-Detection outperforms the strongest baseline (diffusion model Wide ResNet) by 9.35%. The reason that detection based ensembles (GaME-Detection) may perform so well is due to the adversarial label in this voting scheme. For example, in the diffusion model Wide ResNet, the attacker only needs to attack a single model. In the Uniform-Hard Ensemble, the attacker is successful as long as the true class label does not receive the majority of votes. However, for the attacker to successfully attack a GaME-Detection ensemble, one class must receive more votes than any other class. Any tie in voting results in the ensemble producing an adversarial label, and the attack fails in this case. In short, the GaME-Detection framework creates more stringent requirements under which an attacker must operate. Hence GaME-Detection is more robust to adversarial attacks than any other baseline or single defense, as demonstrated by our experimental results.

C. ADAPTIVE GaME EXPERIMENTS (Ada-GaME)

A natural extension to the standard GaME framework is to consider an *adaptive attacker* that seeks to maximize the success of their attacks on a per sample basis. The strength of such an attacker lies in the fact that they can adaptively select which attack performs best not by averaging over inputs,

TABLE 8. (CIFAR-10) Baseline performance of each uniform defense. Uniform-Soft corresponds to soft label voting where softmax values from each model are added together. Uniform-Hard corresponds to hard label voting where only the class label is used in voting. The uniform defense with the highest W-Robust value is given in bold.

	Uniform-Soft Ensemble			Uniform-Hard Ensemble		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	55.00%	92.30%	73.65%	55.10%	92.30%	73.70%
2	54.40%	94.50%	74.45%	54.50%	92.60%	73.55%
3	55.90%	95.90%	75.90%	56.50%	95.00%	75.75%
4	56.30%	96.60%	76.45%	57.20%	95.90%	76.55%
5	56.60%	96.40%	76.50%	57.70%	96.30%	77.00%
6	56.6%	96.80%	76.70%	57.90%	96.50%	77.20%
7	56.70%	96.20%	76.45%	57.90%	96.60%	**77.25%**
8	56.70%	96.50%	76.60%	57.90%	96.60%	**77.25%**

TABLE 9. (Tiny-ImageNet) Baseline performance of each uniform defense. Uniform-Soft corresponds to soft label voting where softmax values from each model are added together. Uniform-Hard corresponds to hard label voting where only the class label is used in voting. The uniform defense with the highest W-Robust value is given in bold.

	Uniform-Soft Ensemble			Uniform-Hard Ensemble		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	26.00%	58.80%	42.40%	26.00%	58.80%	42.40%
2	20.70%	66.20%	43.45%	24.90%	59.10%	42.00%
3	18.80%	69.60%	44.20%	26.10%	62.10%	44.10%
4	18.20%	71.00%	44.60%	26.10%	63.70%	44.90%
5	18.00%	71.30%	44.65%	26.10%	64.10%	**45.10%**

TABLE 10. (CIFAR-10) Performance of GaME with either soft label or hard label voting. The GaME defense with the highest W-Robust value is given in bold.

	GaME-Soft			GaME-Hard		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	71.20%	96.10%	83.65%	71.20%	96.10%	83.65%
2	67.20%	96.40%	81.80%	71.80%	96.20%	84.00%
3	67.00%	96.30%	81.65%	72.10%	96.30%	84.20%
4	67.30%	96.20%	81.75%	72.00%	96.80%	84.40%
5	71.70%	96.80%	84.25%	72.40%	96.80%	**84.60%**
6	71.70%	96.80%	84.25%	72.40%	96.80%	**84.60%**
7	71.70%	96.80%	84.25%	72.40%	96.80%	**84.60%**
8	71.70%	96.80%	84.25%	72.40%	96.80%	**84.60%**

TABLE 11. (Tiny-ImageNet) Performance of GaME with either soft label or hard label voting. The GaME defense with the highest W-Robust value is given in bold.

	GaME-Soft			GaME-Hard		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	28.80%	62.10%	45.45%	28.80%	62.10%	45.45%
2	27.70%	64.70%	46.20%	29.10%	62.30%	45.70%
3	26.10%	67.00%	46.55%	26.10%	62.10%	44.10%
4	26.40%	68.30%	47.35%	28.70%	68.30%	48.50%
5	26.20%	68.70%	47.45%	30.60%	68.00%	**49.30%**

TABLE 12. Performance of GaME with adversarial detection for CIFAR-10 and Tiny-ImageNet. For CIFAR-10 the highest W-Robust is all defense ensembles with $n \geq 2$. For Tiny-ImageNet the highest W-Robust is all defense ensembles with $n \geq 4$.

	(CIFAR-10) GaME-Detection			(Tiny-ImageNet) GaME-Detection		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	71.20%	96.20%	83.70%	26.00%	62.10%	44.05%
2	88.00%	89.10%	88.55%	53.30%	57.20%	55.25%
3	88.00%	89.10%	88.55%	54.10%	57.20%	55.65%
4	88.00%	89.10%	88.55%	56.00%	57.20%	56.60%
5	88.00%	89.10%	88.55%	56.00%	57.20%	56.60%
6	88.00%	89.10%	88.55%			
7	88.00%	89.10%	88.55%			
8	88.00%	89.10%	88.55%			

but by select the best attack one input at a time. Under such a threat model the attacker creates multiple adversarial examples per clean sample and choose one which has the highest attack success rate. When we analyze the GaME

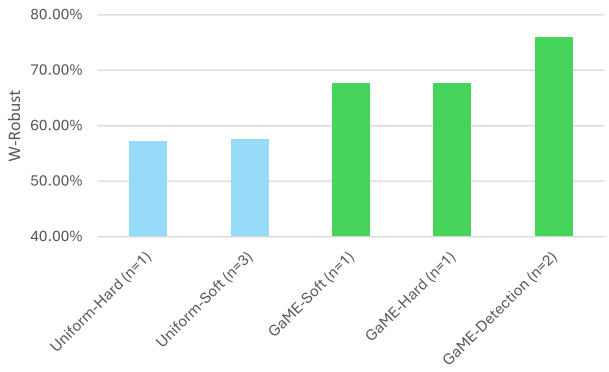

FIGURE 5. (CIFAR-10) Comparison of different baselines and Ada-GaME ensembles under an *adaptive attacker*. For each baseline (Uniform-Soft or Uniform-Hard) and each GaME ensemble we plot the highest W-Robust achieved across all possible values of n. Numerical results can be found in Tables 13and 15.

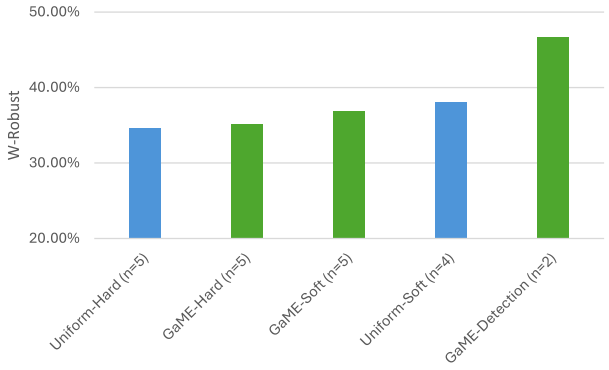

FIGURE 6. (Tiny-ImageNet) Comparison of different baselines and Ada-GaME ensembles under an *adaptive attacker*. For each baseline (Uniform-Soft or Uniform-Hard) and each GaME ensemble we plot the highest W-Robust achieved across all possible values of n. Numerical results can be found in Tables 14and 16.

framework under this setup, we refer to this as Ada-GaME. In order to understand how effective ensembles defenses are under such a threat model, we first develop and analyze uniform ensembles. We then compare these ensembles to Ada-GaME based ensembles.

1) ADAPTIVE ATTACKER BASELINE EXPERIMENTS

In Table 13 and Table 14 we give results for a uniform ensemble defense under attack by an adaptive attacker. The two baselines we use for comparison are Uniform-Soft (soft label voting) and Uniform-Hard (hard label voting). In the interest of fairness, we exclude single defense baselines in this subsection. This is due to fact that these types of defenses are not typically designed or evaluated under this type of adversary.

For CIFAR-10, the highest W-Robust baseline is 57.55% for Uniform-Soft with $n = 3$. For Tiny-ImageNet the the highest W-Robust baseline is 38.05% for Uniform-Soft with $n = 4$. It is worth noting that under a non-adaptive attacker, the Uniform-Hard ensemble baseline performed better (see subsection VII-B). However, when the attacker can adaptively select attacks, Uniform-Soft is more robust. However, both Uniform-Hard and Uniform-Soft perform very similarly. For example, there is less than a 4% difference between the best Uniform-Hard and Uniform-Soft W-Robust value for Tiny-ImageNet. As opposed to a specific defense trend, this indicates the adaptive attacker is strong enough to the point that there is marginal difference between these two baseline techniques.

2) Ada-GaME ENSEMBLES VERSUS BASELINES ANALYSIS

Similar to GaME, we see a significant improvement of Ada-GaME over the uniform defense in terms of robustness against adaptive attacks. In Figure 15 and 16 the W-Robust values for CIFAR-10 and Tiny-ImageNet are shown for the different ensemble baselines and Ada-GaME frameworks. We test three version of Ada-GaME, the soft label voting, the hard label voting and the detection based voting scheme. Just like for the non-adaptive attacker GaME framework, we can clearly see that the detection based approach is most effective. In this set of experiments GaME-Detection has the highest W-Robust measure with 76% for CIFAR-10 and 46.75% for Tiny-ImageNet. For CIFAR-10 the GaME-Detection framework is 18.45% better than the best baseline (Uniform-Hard). Likewise for Tiny-ImageNet the GaME-Detection ensemble is 12.1% better.

There are several other interesting findings that our experiments reveal. In the standard GaME framework a general trend was that increasing n, increased robustness. When the attacker is given the ability to be adaptive (Ada-GaME), we see that increasing n often yields more significant performance improvements. For example, for CIFAR-10 GaME-Hard, the difference between $n = 1$ and $n = 8$ for robustness (not W-Robust) was only 0.5%. In Ada-GaME the difference between $n = 1$ and $n = 8$ for GaME-Hard is 9.9%. It is also worth noting that only certain versions of GaME do better than the baselines under the adaptive attacker. In Figure 16 we can clearly see that Uniform-Soft with $n = 4$ outperforms all other GaME frameworks, except for GaME-Detection. This result highlights that only GaME-Detection reliably outperforms other methods across all datasets and attackers that we tested.

D. STUDY OF COMPUTATIONAL COST

As the value of n increases in a GaME$_n$ defense, the number of possible choices for the defender grows in accordance to the binomial coefficient, $\binom{|D|}{n}$, where D is the set of all single model classifiers. Due to this, we provide a brief study on the effect of n on the computational complexity of creating a GaME$_n$ defense.

The computation time for forming the game-matrix largely depends on the time needed to compute the predictions of each of the defenses for each of the attacks. This is due to the fact that each defense has a different model complexity. Thus, rather than recomputing defense predictions when evaluating each ensemble, we can instead run each set of samples, s_i,

TABLE 13. (CIFAR-10) Baseline performance for an *adaptive attacker* for each uniform defense. Uniform-Soft corresponds to soft label voting where softmax values from each model are added together. Uniform-Hard corresponds to hard label voting where only the class label is used in voting. The uniform defense with the highest W-Robust value is given in bold.

	Uniform-Soft Ensemble			Uniform-Hard Ensemble		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	22.10%	92.30%	57.20%	22.10%	92.30%	57.20%
2	20.30%	94.50%	57.40%	17.60%	92.60%	55.10%
3	19.20%	95.90%	**57.55%**	18.80%	95.00%	56.90%
4	18.30%	96.60%	57.45%	17.40%	95.90%	56.65%
5	17.90%	96.40%	57.15%	17.20%	96.30%	56.75%
6	17.50%	96.80%	57.15%	16.80%	96.50%	56.65%
7	17.50%	96.20%	56.85%	16.70%	96.60%	56.65%
8	17.50%	96.50%	57.00%	16.70%	96.60%	56.65%

TABLE 14. (Tiny-ImageNet) Baseline performance for an *adaptive attacker* for each uniform defense. Uniform-Soft corresponds to soft label voting where softmax values from each model are added together. Uniform-Hard corresponds to hard label voting where only the class label is used in voting. The uniform defense with the highest W-Robust value is given in bold.

	Uniform-Soft Ensemble			Uniform-Hard Ensemble		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	7.20%	58.80%	33.00%	7.20%	58.80%	33.00%
2	5.10%	66.20%	35.65%	5.30%	59.10%	32.20%
3	5.10%	69.60%	37.35%	5.30%	62.10%	33.70%
4	5.10%	71.00%	**38.05%**	5.30%	63.70%	34.50%
5	3.60%	71.30%	37.45%	5.20%	64.10%	34.65%

TABLE 15. (CIFAR-10) Performance of different GaME frameworks for an *adaptive attacker*. The Ada-GaME defense ensembles with the highest W-Robust value is given in bold.

	GaME-Soft			GaME-Hard			GaME-Detection		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	39.30%	96.10%	67.70%	39.30%	96.10%	67.70%	39.30%	96.20%	67.75%
2	37.00%	96.40%	66.70%	37.00%	96.20%	66.60%	62.90%	89.10%	**76.00%**
3	24.50%	96.30%	60.40%	24.80%	96.30%	60.55%	62.90%	89.10%	**76.00%**
4	24.60%	96.20%	60.40%	24.90%	96.80%	60.85%	62.90%	89.10%	**76.00%**
5	31.60%	96.80%	64.20%	29.40%	96.80%	63.10%	62.90%	89.10%	**76.00%**
6	31.60%	96.80%	64.20%	29.40%	96.80%	63.10%	62.90%	89.10%	**76.00%**
7	31.60%	96.80%	64.20%	29.40%	96.80%	63.10%	62.90%	89.10%	**76.00%**
8	31.60%	96.80%	64.20%	29.40%	96.80%	63.10%	62.90%	89.10%	**76.00%**

TABLE 16. (Tiny-ImageNet) Performance of different GaME frameworks for an *adaptive attacker*. The Ada-GaME defense ensembles with the highest W-Robust value is given in bold.

	GaME-Soft			GaME-Hard			GaME-Detection		
n	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust	Robustness	Clean	W-Robust
1	8.00%	59.90%	33.95%	8.00%	59.90%	33.95%	8.00%	59.90%	33.95%
2	7.10%	64.50%	35.80%	6.50%	59.80%	33.15%	45.10%	48.40%	**46.75%**
3	6.80%	65.40%	36.10%	6.20%	62.30%	34.25%	42.20%	50.70%	46.45%
4	6.20%	67.70%	36.95%	6.10%	63.80%	34.95%	39.60%	53.00%	46.30%
5	6.10%	67.80%	36.95%	6.00%	64.40%	35.20%	39.30%	53.20%	46.25%

through each defense $d \in D$ once, receiving output $y_{i,d}$ for each sample, defense pair. To get the robust accuracy of $U \subset D$ when evaluating samples s_i we can substitute $y_{i,d}$ for $d(s_i)$ in the computation of $f^h(s_i, U)$ or $f^s(s_i, U)$ (see Equation 14 and Equation 15). This means that we do not need to perform a number of model evaluations that scales with n.

In Figure 7 we show the computational cost of creating the game matrix and solving the associated linear program for a GaME$_n$ ensemble as a function of n. Model prediction time is not considered in the calculation as it does not depend on n, as explained previously. These experiments were run on a computer with the following specifications: Intel Core i9-10900K CPU @ 3.70GHz, Nvidia RTX3080 12GB, and 64 GB RAM.

The computational cost for the random transform defenses were significantly greater than defenses that make a direct prediction due to the added complexity of transforming the input images. For the BaRT defenses this was mitigated by running the CPU based transformations in parallel on a per-sample basis. The computational cost for evaluating the BiT-ViT Trash is Treasure defense is the highest since it requires running a 13 step PGD attack against a large BiT model for each sample before it is given to the main ViT

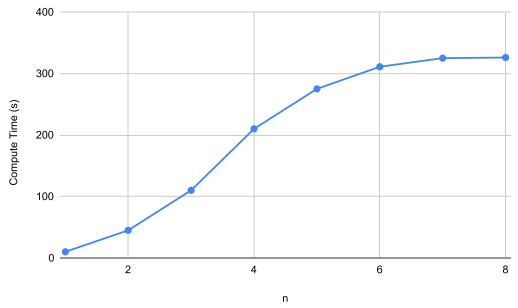

FIGURE 7. (CIFAR-10) Study of the computational cost of the GaME framework. Time cost as a function of *n*, the number of defenses.

classifier. This transformation is also run in parallel on the GPU.

VIII. CONCLUSION

In this paper, we have advanced adversarial machine learning by developing and analyzing a new game theoretic framework. In both the attack and defense parts of the field we make significant contributions. On the attack side, we proposed three new white-box attacks. We developed the Momentum Iterative Method over Expectation (MIME) for attacking single randomized defenses, and we created the Multi-Agent Gradient Expectation attacks (MAGE and MAGE-D) for dealing with a combination of randomized and non-randomized defenses and detectors. Additionally, we are the first to show the transferability of adversarial examples generated by attacks like MIM, APGD, MIME and MAGE on SOTA defenses.

From the defense side, we develop a game theoretic framework to approximate an optimal strategy for adversarial attackers and defenders. The flexibility of this framework allows any newly proposed defense or attack to be easily integrated. Using a set of SOTA attacks and defenses we demonstrate that our game theoretic framework can create a compositional defense that is 18% more robust than the best single model defense for CIFAR-10 and is 27% more robust on Tiny-ImageNet. In terms of W-Robust, our GaME-Detection ensemble is 5.4% and 9.35% better on CIFAR-10 and Tiny-ImageNet than the next best SOTA defense. We further analyze our GaME framework under a stronger adaptive adversary and show that it is superior to ensembles that use naive, uniform selection strategies. Overall, our work yields a new framework for optimizing multi-defense voting strategies to obtain higher levels of robustness.

APPENDIX

A. ADDITIONAL ATTACK EXPERIMENTS

MAGE-D makes significant improvements over MAGE when attacking defenses utilizing detectors as shown in Table 17. For instance, when attacking the Tiny ImageNet detector, 83% of the samples generated by MAGE are either correctly classified or labeled as adversarial by the detector, but with MAGE-D the detector's robustness drops to 0%. This shows that MAGE-D is able to effectively break detector defense architectures by running two additional targeted attacks on image.

TABLE 17. (CIFAR-10) Comparison between MAGE and MAGE-D when attacking detector models utilizing vanilla voting models. The MAGE and MAGE-D columns represent the robust accuracy of the detector defense when evaluating samples generated by the MAGE and MAGE-D defenses respectively.

Dataset	Model1	Model2	Clean	MAGE	MAGE-D
CIFAR-10	ResNet-164	BiT-M-50x1	97.00%	60.40%	10.40%
Tiny ImageNet	ViT-L-16	BiT-M-50x1	69.70%	85.30%	0.00%

TABLE 18. (CIFAR-10) Testing the effectiveness of MAGE against a defense with 3 vanilla classifiers. Each column represents the robust accuracy of the respective model or defense when evaluating the samples generated by MAGE when attacking all three models. The last column represents the robust accuracy of an ensemble defense utilizing all three models and the soft label voting scheme.

Attack	R	V	SB	R+V+SB
M(R,V,SB)	63.3%	31.6%	45.5%	54.6%

For our implementation of GaME$_n$ we performed MAGE attacks against 2 model ensembles. This was for two reasons: MAGE can have a high computational cost such that attacking size 3 ensembles increases the total number of experiments needed exponentially, and second, MAGE does not scale well to more than 2 defenses. We provide an empirical evidence of this below in Table 18 where we attack a vanilla ResNet-164 (R), vanilla ViT-L-16 (V), and a vanilla SNN model trained using back propagation.

B. LINEAR PROGRAMS FOR SOLVING GaME

We present the explicit linear program for solving GaME as the attacker. Let $O_A = (\lambda^A_{a_1}, \cdots \lambda^A_{a_{|A|}}, r^*)$ be the row vector containing the elements of λ^A and r^* and $R_{d,a} = r_{d,a}$ by the payoff matrix for p_D. Additionally let $\hat{0}$ denote the zero vector. The attacker must then solve the following linear program:

$$\begin{aligned} &\max \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} O_A^T \\ &\text{Subject to: } \begin{pmatrix} -r_{d_1,a_1} & -r_{d_1,a_2} & \cdots & 1 \\ -r_{d_2,a_1} & -r_{d_2,a_2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} O_A^T \leq \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \\ &\text{and: } O_A \geq \hat{0} \end{aligned} \tag{35}$$

Next, we show the explicit linear program for solving GaME as the defender. For convenience let $O_D = (\lambda^D_{d_1}, \cdots \lambda^D_{d_{|D|}}, r^*)$ be the row vector containing the elements of λ^D and r^*. The

defender must solve the following linear program:

$$\begin{aligned} &\max \ \begin{pmatrix} 0 & \cdots & 0 & 1 \end{pmatrix} O_D^T \\ &\text{Subject to:} \ \begin{pmatrix} -r_{d_1,a_1} & -r_{d_2,a_1} & \cdots & 1 \\ -r_{d_1,a_2} & -r_{d_2,a_2} & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 0 \end{pmatrix} O_D^T \leq \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix} \\ &\text{and: } O_D \geq \hat{0} \end{aligned} \tag{36}$$

Due to the nature of the dual problem in linear programming, solving this problem will result in the same value for r^* as was found in the primal problem presented in the main body of the paper.

ACKNOWLEDGMENT

The authors would like to acknowledge each of them for their work in contributing to the article. No artificial intelligence (AI) was used in producing the writing for this article.

REFERENCES

[1] S. Abnar and W. Zuidema, "Quantifying attention flow in transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 4190–4197.
[2] A. Araujo, L. Meunier, R. Pinot, and B. Négrevergne, "Advocating for multiple defense strategies against adversarial examples," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2020, pp. 165–177.
[3] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing robust adversarial examples," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 284–293.
[4] M.-F. Balcan, R. Pukdee, P. Ravikumar, and H. Zhang, "Nash equilibria and pitfalls of adversarial training in adversarial robustness games," 2022, *arXiv:2210.12606*.
[5] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," 2019, *arXiv:1902.06705*.
[6] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 3–14.
[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.
[8] P. Chen, Y. Sharma, H. Zhang, J. Yi, and C. Hsieh, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, 2018.
[9] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2206–2216.
[10] J. Deng, A. Palmer, R. Mahmood, E. Rathbun, J. Bi, K. Mahmood, and D. Aguiar, "Distilling adversarial robustness using heterogeneous teachers," 2024, *arXiv:2402.15586*.
[11] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.
[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16×16 words: Transformers for image recognition at scale," in *Proc. Int. Conf. Learn. Represent.*, 2020.
[13] W. Fang, Z. Yu, Y. Chen, T. Huang, T. Masquelier, and Y. Tian, "Deep residual learning in spiking neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 21056–21069.
[14] S. G. Finlayson, J. D. Bowers, J. Ito, J. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, 2019.
[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.
[16] N. M. Gürel, X. Qi, L. Rimanić, C. Zhang, and B. Li, "Knowledge enhanced machine learning pipeline against diverse adversarial attacks," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 3976–3987.
[17] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[18] J. Ho, A. N. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 6840–6851.
[19] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (BiT): General visual representation learning," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 491–507.
[20] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 (Canadian institute for advanced research)," Tech. Rep., 2010.
[21] A. Krizhevsky, V. Nair, and G. Hinton, "CIFAR-10 (Canadian institute for advanced research)," Tech. Rep., 2009.
[22] T. Le, A. T. Bui, H. Zhao, P. Montague, Q. Tran, and D. Phung, "On global-view based defense via adversarial attack and defense risk guaranteed bounds," in *Proc. Mach. Learn. Res.*, vol. 151, Mar. 2022, pp. 11438–11460.
[23] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," Tech. Rep., 2015, vol. 7, no. 7, p. 3.
[24] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," 2016, *arXiv:1611.02770*.
[25] A. Mądry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–11.
[26] K. Mahmood, D. Gurevin, M. van Dijk, and P. H. Nguyen, "Beware the black-box: On the robustness of recent defenses to adversarial examples," *Entropy*, vol. 23, no. 10, p. 1359, Oct. 2021.
[27] K. Mahmood, R. Mahmood, and M. van Dijk, "On the robustness of vision transformers to adversarial examples," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 7818–7827.
[28] P. Maini, E. Wong, and J. Z. Kolter, "Adversarial robustness against the union of multiple perturbation models," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6640–6650.
[29] L. Meunier, M. Scetbon, R. Pinot, J. Atif, and Y. Chevaleyre, "Mixed Nash equilibria in the adversarial examples game," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 7677–7687.
[30] J. Nash, "Non-cooperative games," *Ann. Math.*, vol. 54, no. 2, pp. 286–295, 1951.
[31] A. Pal and R. Vidal, "A game theoretic analysis of additive adversarial attacks and defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 1345–1355.
[32] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving adversarial robustness via promoting ensemble diversity," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4970–4979.
[33] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," 2016, *arXiv:1605.07277*.
[34] R. Pinot, R. Ettedgui, G. Rizk, Y. Chevaleyre, and J. Atif, "Randomization matters how to defend against strong adversarial attacks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 7717–7727.
[35] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 2, pp. 998–1026, 2nd Quart., 2020.
[36] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6521–6530.
[37] N. Rathi and K. Roy, "DIET-SNN: A low-latency spiking neural network with direct input encoding and leakage and threshold optimization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 6, pp. 3174–3182, Jun. 2023.
[38] L. Schmidt, S. Santurkar, D. Tsipras, K. Talwar, and A. Mądry, "Adversarially robust generalization requires more data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018.
[39] V. Sehwag, S. Mahloujifar, T. Handina, S. Dai, C. Xiang, M. Chiang, and P. Mittal, "Robust learning meets generative models: Can proxy distributions improve adversarial robustness?" 2021, *arXiv:2104.09425*.
[40] S. Sengupta, T. Chakraborti, and S. Kambhampati, "MTDeep: Boosting the security of deep neural nets against adversarial attacks with moving target defense," in *Proc. 32nd AAAI Conf. Artif. Intell. Workshops*, 2018, pp. 1–11.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[42] C. Sitawarin, Z. Golan-Strieb, and D. Wagner, "Demystifying the adversarial robustness of random transformation defenses," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 20232–20252.
[43] D. Stutz, M. Hein, and B. Schiele, "Disentangling adversarial robustness and generalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 6969–6980.
[44] F. Tramèr, N. Carlini, W. Brendel, and A. Mądry, "On adaptive attacks to adversarial example defenses," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020.
[45] J. Wang, Z. Lyu, D. Lin, B. Dai, and H. Fu, "Guided diffusion model for adversarial purification," 2022, *arXiv:2205.14969*.
[46] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *Proc. Int. Conf. Learn. Represent.*, 2020, pp. 1–12.
[47] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan, "Better diffusion models further improve adversarial training," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 36246–36263.
[48] C. Xiao and C. Zheng, "One man's trash is another man's treasure: Resisting adversarial examples by adversarial examples," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 412–421.
[49] N. Xu, K. Mahmood, H. Fang, E. Rathbun, C. Ding, and W. Wen, "Attacking the spike: On the transferability and security of spiking neural networks to adversarial examples," 2022, *arXiv:2209.03358*.
[50] G.-Q. Zeng, J.-M. Shao, K.-D. Lu, G.-G. Geng, and J. Weng, "Automated federated learning-based adversarial attack and defence in industrial control systems," *IET Cyber-Syst. Robot.*, vol. 6, no. 2, Jun. 2024, Art. no. e12117.
[51] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7472–7482.
[52] J. Zhang, X. Xu, B. Han, G. Niu, L. Cui, M. Sugiyama, and M. Kankanhalli, "Attacks which do not kill training make adversarial learning stronger," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 11278–11287.
[53] S. Zhao, X. Wang, and X. Wei, "Mitigating accuracy-robustness trade-off via balanced multi-teacher adversarial distillation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9338–9352, Dec. 2024.

KALEEL MAHMOOD received the M.S. degree in electrical engineering and the Ph.D. degree in computer science from the University of Connecticut, in 2016 and 2022, respectively. He is currently an Assistant Professor with the Department of Computer Science and Statistics, The University of Rhode Island. His research spans a broad range of topics, including adversarial machine learning, security, image processing, and natural language processing.

ETHAN RATHBUN received the B.S. degree in computer science and the B.A. degree in mathematics from the University of Connecticut, in 2022. He is currently pursuing the Ph.D. degree in computer science with Northeastern University, where he studies the intersection of reinforcement learning and cybersecurity. His current research interest includes the capabilities of training time attacks against reinforcement learning algorithms.

RONAK SAHU received the B.S.E. degree in computer science and engineering and the B.S. degree in applied mathematics from the University of Connecticut, in 2023. He is currently pursuing the Ph.D. degree in computer science and engineering. His research interests include machine learning and cybersecurity.

MARTEN VAN DIJK (Fellow, IEEE) leads the Computer Security Group, CWI, The Netherlands, with more than 20 years of experience in both industry (Philips Research and RSA Laboratories) and academia (MIT, University of Connecticut, and currently Vrije Universiteit van Amsterdam). His work has been recognized by the IEEE CS Edward J. McCluskey Technical Achievement Award, in 2023, and the A. Richard Newton Technical Impact Award in Electronic Design Automation, in 2015, and has received several best and test-of-time article awards.

SOHAIB AHMAD received the B.S. degree in computer science from the National University of Science and Technology, Pakistan, and the Ph.D. degree in computer science from the University of Connecticut, in 2022. He is currently a Senior Data Scientist with the Department of Cyber AI and Innovation, Visa Inc. His research interests include biometrics and the implication of security on machine learning models.

CAIWEN DING (Member, IEEE) received the Ph.D. degree from Northeastern University, in 2019. He is currently an Associate Professor with the Department of Computer Science and Engineering, University of Minnesota, Minneapolis. His research interests include algorithm-system co-design of machine learning/artificial intelligence, computer architecture, and heterogeneous computing.