Check for updates

Received: 30 June 2025 | Revised: 14 January 2026 | Accepted: 9 February 2026

DOI: 10.4218/etrij.2025-0264

ORIGINAL ARTICLE

ETRI Journal WILEY

# Fast adversarial robustness recovery from model parameter exposure through Hopfield-type neural ordinary differential equations

Yu-Hyun Shin[1] | Kimin Yun[2,3] | Yuseok Bae[2]

[1]Korea Automotive Technology Institute (KATECH), Cheonan, Republic of Korea

[2]Visual Intelligence Lab., Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea

[3]University of Science and Technology, Daejeon, Republic of Korea

**Correspondence**
Kimin Yun, Visual Intelligence Lab., Electronics and Telecommunications Research Institute, Daejeon, Republic of Korea.
Email: kimin.yun@etri.re.kr

**Funding information**
This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP), Grant/Award Number: RS-2022-II220124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities)

**Abstract**

Model parameter leakage transforms black-box settings into near-white-box threats, enabling the insertion of highly transferable adversarial examples through exposed model internals. Defending against such attacks requires fast, robust, and minimally invasive adaptation methods. To address this challenge, we introduce a lightweight plug-in defense module that refines corrupted latent features while preserving a frozen backbone model. Our approach leverages ordinary neural differential equations with a Hopfield-type architecture, which is designed to purify latent features with strong stability properties. The key innovation of our model is a channel gain normalization technique that analytically binds the Lipschitz constant, thereby improving training stability and robustness. Notably, our method requires only 5% of the original training data to outperform standard adversarial training on both clean and adversarial samples, achieving 84.8% robust accuracy on CIFAR-10 (+6.2% over adversarial fine-tuning), while maintaining a clean accuracy of 90.4%. Furthermore, our model effectively adapts to repeated model leakages and memory-aware attacks by supporting rapid reinitialization. Extensive experiments on the CIFAR-10 and CIFAR-100 datasets demonstrate the effectiveness of the proposed approach. This work represents a practical step toward fast, computationally adaptive defenses against model leakage attacks, which are becoming increasingly common in modern machine learning environments.

**KEYWORDS**
adversarial attack, Hopfield-type network, image classification, model leakage, neural ordinary differential equation

## 1 | INTRODUCTION

Deep learning has led to remarkable advances on various visual tasks [1-3]. However, the susceptibility of deep learning systems to adversarial examples is a critical security concern. Adversarial attacks involve the insertion of carefully crafted perturbations into input data, which are often imperceptible to humans and may lead

Yu-Hyun Shin and Kimin Yun should be considered joint first authors.

This is an Open Access article distributed under the term of Korea Open Government License (KOGL) Type 4: Source Indication + Commercial Use Prohibition + Change Prohibition (http://www.kogl.or.kr/info/licenseTypeEn.do).
1225-6463/$ © 2026 ETRI

to incorrect predictions by deep neural networks (DNNs). This vulnerability has been widely observed across diverse DNN architectures and has become a major focus area owing to its serious implications for the reliability and security of real-world applications.

Adversarial attacks are generally categorized according to the attacker's knowledge of the target model. *White-box* attacks assume full access to model parameters, whereas *black-box* attacks rely on transferability or iterative queries, without internal access [4-7]. In practice, many real-world scenarios fall between these extremes, and when trained model parameters are accidentally or intentionally exposed, an attacker can gain partial white-box knowledge. This *model leakage* effectively transforms transfer-based attacks into near-white-box threats, creating a unique challenge for existing defense measures, which are either too computationally heavy or too rigid to adapt quickly.

Furthermore, the increasing use of large-scale foundation models introduces additional complexity. Although such models are generally robust against natural data variations, they remain vulnerable to adversarial perturbations [8] that can be exploited in transfer attacks [9,10]. In addition to requiring substantial computational resources, fine-tuning these models to defend against adversarial attacks presents a critical dilemma: Updating models to defend against transfer attacks can inadvertently degrade their performance on clean inputs [11].

Despite extensive research, current defense mechanisms face fundamental limitations when addressing model leakage scenarios. Adversarial training (AT) [12,13], which is effective against known attacks, requires retraining an entire model using adversarial examples. This process requires substantial computational resources and often degrades clean accuracy by 10% to 15% [11]. Additionally, full retraining is prohibitively expensive for large-scale foundation models. Input preprocessing methods [14] and architectural modifications [15] offer model-agnostic solutions but remain vulnerable to adaptive attacks that exploit knowledge of defense mechanisms. Most critically, existing modular defense approaches [16] still require updating entire models, which limits their practical applicability when rapid adaptation is required following repeated-leakage incidents.

To address these limitations, we propose a fundamentally different defense paradigm based on a simple principle. Rather than retraining an entire model, we insert a lightweight "feature purifier" that operates exclusively in the latent space of the early network layers. This approach was motivated by two key observations. First, adversarial perturbations in shallow layers have not yet fully propagated into decision-critical features, and they remain in a malleable state, where targeted intervention can redirect them toward natural data manifolds. Second, by freezing all pretrained weights and training only the inserted module, we can preserve the model's original knowledge while enabling rapid adaptation using only 5% of the training data.

The core challenge lies in the design of a lightweight and stable purification module. Stability analysis has become an important topic in continuous-time neural models, including neural ordinary differential equations (ODEs) and more general formulations such as fractional-order neural networks with Mittag-Leffler stability [17]. A purification module must effectively neutralize adversarial noise without introducing architectural complexity, which would negate its computational advantages. To this end, we leverage neural ODEs [18], which model continuous transformations in the latent space and preserve topological structures through homeomorphic mappings that smoothly deform input features without destroying essential relationships. This property enables the proposed module to redirect adversarial perturbations away from critical directions while maintaining clean feature integrity.

Among the various available ODE formulations, we adopted Hopfield-type dynamics [19] for two critical reasons. First, Hopfield networks exhibit energy-minimizing behaviors, as they naturally evolve toward stable attractor states, rendering them well-suited for "pulling" corrupted features back toward clean data manifolds. Second, unlike general neural ODEs, Hopfield-type ODEs provide analytically bounded Lipschitz continuity, ensuring stable gradient flows during training on adversarial data. This theoretical guarantee is essential for preventing gradient explosions when the module encounters strong perturbations during optimization.

Our implementation incorporates three key techniques to enable rapid and stable adaptation. First, we introduce channel gain normalization (CGN), which separates the direction and intensity of shared weights in the module. This design provides enhanced gradient control compared with conventional weight normalization, enabling more stable ODE dynamics during training. Second, we adopt a reinitialization strategy that supports rapid adaptation to new attack patterns following repeated model leakage incidents. Third, we apply learning rate rewarming to support efficient optimization while preserving clean input performance.

The main contributions of this study can be summarized as follows.

- We introduce a lightweight Hopfield-type ODE module that is designed to defend against adversarial attacks in model leakage scenarios. To handle repeated-leakage events, we incorporate reinitialization and learning rate rewarming strategies that support rapid and effective adaptation.

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

- We propose CGN as a method tailored to Hopfield-type ODEs, which improves gradient stability by decoupling the direction and magnitude of shared weights, securing more reliable training dynamics.
- We demonstrate that freezing pretrained model parameters and training only the inserted ODE module yields improved robustness against adversarial examples from leaked models while preserving clean accuracy.

Although our primary focus is defending against transfer attacks, given their practical relevance in model leakage scenarios, our method is also effective against white-box attacks when sufficient computational resources are available. Extensive experiments on various datasets and architectures demonstrate the effectiveness of the proposed approach for restoring adversarial robustness following model leakage.

## 2 | RELATED WORKS

### 2.1 | Neural ODEs

Traditional DNNs are composed of discrete layers that transform data sequentially. The neural ODEs introduced by Chen [18] reinterpret this process as a continuous transformation over time. Rather than stacking layers, neural ODEs model the evolution of hidden states as solutions to ODEs parameterized by neural networks. This approach is naturally motivated by architectures such as ResNet [20], where each residual block computes

$$x_{t+1} = x_t + \mathrm{F}(x_t), \tag{1}$$

where $\mathrm{F}(x_t)$ represents a learnable function such as a convolution layer and $x_t$ denotes the output of the previous $t$ th layer. This structure resembles the forward Euler discretization of the ODE $\frac{dx}{dt} = \mathrm{F}(x)$. When representing the learnable parameters of $F$ as $\theta$, neural ODEs generalize this structure by replacing discrete steps with continuous dynamics as follows:

$$\frac{dx}{dt} = \mathrm{F}(x, t; \theta), \tag{2}$$

which enables the model to evolve into a hidden state over continuous intervals.

Neural ODEs generally preserve input topological structures because continuous dynamics cannot easily create or destroy features. Although Dupont and others [21] considered this characteristic as a limitation in expressivity and proposed augmented dimensions to address this issue, topological stability can actually enhance robustness. Empirical studies have suggested that neural ODEs are inherently more stable under input perturbations compared with their discrete counterparts [22,23]. Several extensions have been proposed to improve robustness further. Neural stochastic differential equations [24] inject noise during integration for regularization. Kang and others [25] explored the dynamics converging toward stable solutions under input shifts, while Yang and others [26] investigated the damping mechanisms in discrete ODE formulations for adversarial robustness. Purohit [27] proposed an orthogonal convolution to constrain the Lipschitz constants of neural ODEs. More recently, the stability properties of ODE-based neural architectures have been investigated systematically to enhance their robustness. For example, Luo and others [28] analyzed stable mappings in neural and nonautonomous ODEs, whereas Yang and others [29] proposed stability-constrained nmODE models to defend against adversarial perturbations.

Although these studies established the robustness of neural ODEs, they did not address the specific challenge of rapid adaptation following model leakage. Additionally, the existing Lipschitz-constrained methods either lack analytical bounds for multi-channel convolutions or require architectural changes that are incompatible with plug-in deployment. Our study bridges this gap by providing channel-wise normalization with provable stability guarantees, thereby enabling rapid robustness recovery without modifying the backbone model.

### 2.2 | Adversarial attacks

Adversarial attacks deliberately perturb input data, causing DNNs to make incorrect predictions. These perturbations are typically imperceptible to humans but can significantly degrade model performance, raising security concerns for AI systems in critical applications [30]. Such attacks are generally categorized into white-box and black-box attacks depending on the attacker's access level to the target model.

In white-box settings, attackers have full knowledge regarding the model architecture and parameters, enabling the application of powerful gradient-based methods such as the fast gradient sign method (FGSM) [12], projected gradient descent (PGD) [13], and Carlini–Wagner (CW) attacks [31], which optimize adversarial inputs to maximize prediction loss. Although effective, these attacks are rarely feasible in real-world applications, where internal model details are typically inaccessible [32].

Black-box attacks address this limitation by eliminating the requirement for model transparency. Such attacks

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

fall into two primary subtypes: query- and transfer-based attacks. Query-based attacks estimate gradient information through repeated interactions with the target model [33]. However, they often require an impractical number of queries, rendering them slow and easily detectable [34]. Transfer-based attacks exploit adversarial transferability [35], where inputs designed to deceive one model can also fool another model. This property makes transfer attacks particularly concerning because they can be mounted without direct access to the target model, using only a substitute model for crafting attacks.

The model leakage scenario we address represents a hybrid threat, where attackers gain white-box knowledge through parameter exposure but must still rely on transfer attacks for evaluation because they cannot access the updated target model in real time. This scenario represents a unique defense challenge. Existing white-box defenses assume that attackers use real-time model parameters, whereas black-box defenses underestimate threats when attackers possess detailed architectural knowledge. Our approach is designed specifically for this intermediate regime.

### 2.3 | Adversarial defenses

Various defense strategies have been developed to improve the robustness of models against adversarial attacks. These methods can be classified into three main categories: training-based defense, input/architectural modifications, and modular approaches.

Training-based defenses modify the learning process to enhance robustness. AT [12,36] is one of the most effective approaches to incorporating adversarial examples during training. However, standard AT requires the generation of adversarial examples at every training step, resulting in two to three times the computational overhead and often converging only after 100+ epochs. Furthermore, the min-max optimization objective frequently leads to overfitting on adversarial examples, degrading clean accuracy by 10% to 15% [11]. Recent analyses have highlighted the intrinsic limitations of AT, including robustness saturation and poor generalization beyond the observed attack distributions [37].

Advanced variants such as TRADES [11] and MART [38] refine this process by implementing custom loss functions that balance natural accuracy and adversarial robustness. However, these methods require full model retraining, which is expensive for large-scale models or repeated-leakage scenarios.

In addition to training-centric approaches, recent studies have explored adversarial robustness from an architectural perspective. Dong and others [39] investigated robust neural architectures with improved resistance to strong adversarial attacks. Related approaches focus on reducing model sensitivity by regulating Lipschitz constants [40,41] with efficient estimation methods for convolutional layers [42,43]. Latent-space regularization strategies such as mixup [44] and intraclass compactness constraints [45] also enforce stability in internal representations during training.

Input-level defenses sanitize data to neutralize adversarial noise before it reaches the model. Denoising approaches [46] align the logits of clean and adversarial inputs, whereas transformations such as JPEG compression, total variance minimization, and bit-depth reduction [14] suppress adversarial signals in a model-agnostic manner. Although these methods are computationally efficient, they are inherently vulnerable to adaptive attacks. Once an adversary knows the specific transformation applied, they can craft attacks robust to that type of preprocessing [47]. Furthermore, aggressive transformations often degrade clean accuracy by removing task-relevant, high-frequency information.

Architectural modifications include channel-level pruning [15,48], which mitigates the effect of neurons prone to adversarial activation [49]. However, pruning-based methods require retraining from scratch to maintain accuracy, which limits their applicability when defending pretrained models.

Recently, modular defense strategies have attracted significant attention as flexible alternatives. Kim and others [16] proposed restoring perturbed latent features using an auxiliary module, although this approach requires full retraining of the base model, limiting its practical applicability. Similarly, Shan and others [50] investigated post-breach defenses under model leakage but focused on general countermeasures rather than rapid robustness recovery. Although these works recognize the value of modular architectures, they still update all model parameters during defense training.

In contrast, our approach freezes the backbone model entirely and updates only the lightweight ODE modules. This design enables 100× faster adaptation while preserving the clean accuracy of the original model. Table 1 summarizes the key trade-offs across the defense categories. Our method uniquely combines modular flexibility with minimal retraining costs, making it particularly suitable for model leakage scenarios in which rapid repeated adaptation is required.

## 3 | PROPOSED METHOD

Our defense method operates based on the principle of early-stage feature purification. As shown in Figure 1,

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

**TABLE 1** Comparison of adversarial defense methods across key dimensions.

| Method | Retraining scope | Epochs needed | Clean Acc. trade-off | Adaptation speed | Inference overhead | Leakage ready |
|---|---|---|---|---|---|---|
| AT [13] | Full model | 100+ | Moderate | Slow | - | × |
| TRADES [11] | Full model | 100+ | Moderate | Slow | - | × |
| Input transform [14] | None | - | Small | Fast | Small | ✓ |
| Pruning [15] | From scratch | 200+ | Moderate | Very slow | - | × |
| Modular [16] | Full model | 50+ | Moderate | Moderate | moderate | × |
| Ours | Module only | 1 | Negligible | Fast | Small | ✓ |

*Note*: Our approach achieves fast adaptation and clean accuracy preservation without full model retraining.

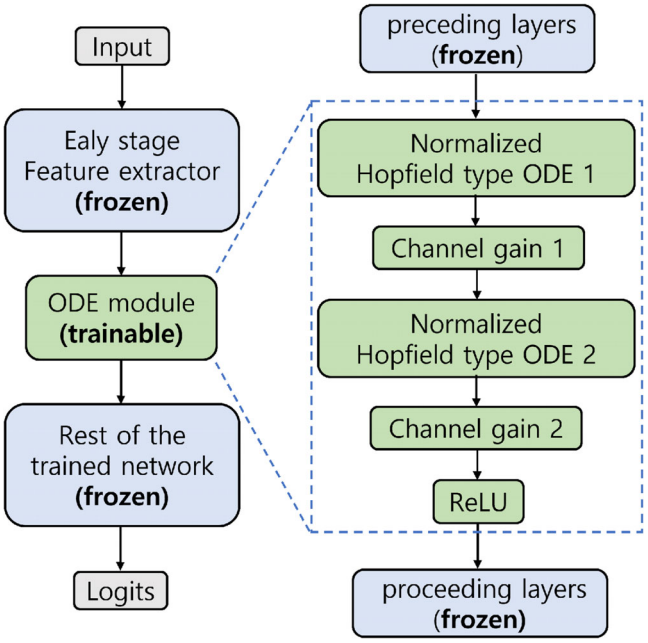


**FIGURE 1** Structure and information flow of the proposed ODE module. *Architecture*: Two sequential channel-gain-normalized Hopfield-type ODEs followed by a rectified linear unit (ReLU), inserted after the first activation of the frozen backbone. *Mechanism*: Input features contaminated with adversarial perturbations (deviating from the clean manifold) are redirected toward stable states through the ODE module. CGN ensures stable dynamics during purification. The corrected features then propagate through deeper layers, preventing the original perturbation from causing misclassification.

adversarial perturbations entering the network through the input layer corrupt feature representations in the shallow convolutional layers. However, during this early stage, these perturbations have not yet propagated sufficiently deep to alter decision-critical features irreversibly. By inserting a Hopfield-type ODE module immediately after the first activation layer, we intercept and correct corrupted features. The module redirects input features back toward a clean data manifold by leveraging energy minimization dynamics before they cascade through deeper layers and cause misclassification.

To enable the rapid robustness recovery of models exposed to parameter leakage, we propose a plug-in module based on Hopfield-type dynamics within an ODE framework. As shown in Figure 1, this module consists of two sequential Hopfield-type ODEs, followed by ReLU activation. The module is inserted after the first activation function of the backbone model. This architecture leverages Hopfield-inspired flows that guide perturbed representations back toward stable configurations. Unlike conventional ODE layers, this mechanism exhibits energy-minimizing behavior, which aids in restoring feature robustness following adversarial corruption.

Although theoretically promising, the training of such modules remains challenging. Shin and Baek [19] applied weight normalization to ODEs; however, we observed that this approach often results in unstable training, including exploding gradients [51]. To address this issue, we introduce CGN, which binds the Lipschitz constant by normalizing the gain of each feature channel during training. This technique mitigates gradient explosions and facilitates stable integration into pretrained models. Crucially, only the ODE module is trained, whereas the backbone weights remain frozen, enabling rapid adaptation without full retraining.

## 3.1 | Weight normalization issue

To formalize the aforementioned limitations, we examine the effectiveness of directly applying weight normalization [52] to Hopfield-type ODEs. Although this operation stabilizes training by decoupling the weight magnitude and direction, its effect diminishes when applied to Hopfield-type ODEs involving channel interactions.

Consider a flattened input feature vector $\mathbf{x} = [x_1, x_2, ..., x_c]^{\mathrm{T}}$, where $x_i$ denotes the $i$ th channel feature of shape $1 \times n^2$ for input feature maps of size $n \times n$. The Hopfield-type ODE is defined as

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

$$\dot{\mathbf{x}}(t) = -\mathbf{W}^{\mathrm{T}}\sigma(\mathbf{W}\mathbf{x}(t)) - \epsilon \cdot \mathbf{x}(t) + \mathrm{F}(\mathbf{x}(0)), \tag{3}$$

where $\sigma(\cdot)$ is the ReLU activation, $\epsilon$ is a damping constant, $\mathrm{F}(\cdot)$ is a $1 \times 1$ convolution followed by ReLU, and $\mathbf{W}$ represents a composite convolutional operator that captures cross-channel interactions.

The convolutional matrix $\mathbf{W}$ can be decomposed into $c \times c$ blocks as follows:

$$\mathbf{W} = \begin{bmatrix} \mathbf{W}_{1,1} & \cdots & \mathbf{W}_{c,1} \\ \vdots & \ddots & \vdots \\ \mathbf{W}_{1,c} & \cdots & \mathbf{W}_{c,c} \end{bmatrix}, \tag{4}$$

where $\mathbf{W}_{i,j}$ represents the $i$ th input's contribution to the $j$ th output channel via convolution. Applying weight normalization directly reparametrizes $\mathbf{W}$ as the product of a gain matrix $\mathbf{G}_{\mathrm{WN}}$ and normalized weight matrix $\mathbf{V}$ as follows:

$$\mathbf{W} = \mathbf{G}_{\mathbf{WN}}\mathbf{V} = \begin{bmatrix} g_1\mathbf{I} & 0 & \cdots & 0 \\ 0 & g_2\mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \cdots & g_c\mathbf{I} \end{bmatrix} \begin{bmatrix} \mathbf{V}_{1,1} & \mathbf{V}_{2,1} & \cdots & \mathbf{V}_{c,1} \\ \mathbf{V}_{1,2} & \mathbf{V}_{2,2} & \cdots & \mathbf{V}_{c,2} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{V}_{1,c} & \mathbf{V}_{2,c} & \cdots & \mathbf{V}_{c,c} \end{bmatrix}, \tag{5}$$

where $g_i$ denotes the gain of the $i$ th channel output. In this setting, the composed term $\mathbf{W}^{\mathrm{T}}\mathbf{W}$ becomes

$$\mathbf{W}^{\mathrm{T}}\mathbf{W} = \begin{bmatrix} \sum_k g_k^2 \mathbf{V}_{1,k}^{\mathrm{T}}\mathbf{V}_{k,1} & \cdots & \sum_k g_k^2 \mathbf{V}_{1,k}^{\mathrm{T}}\mathbf{V}_{k,c} \\ \vdots & \ddots & \vdots \\ \sum_k g_k^2 \mathbf{V}_{c,k}^{\mathrm{T}}\mathbf{V}_{k,1} & \cdots & \sum_k g_k^2 \mathbf{V}_{c,k}^{\mathrm{T}}\mathbf{V}_{k,c} \end{bmatrix}. \tag{6}$$

This result reveals a key issue, namely, that the gain coefficients $g_k$, which are intended to scale each channel independently, become entangled across all output channels. Consequently, the core purpose of weight normalization for channel-wise gain control fails in this multi-channel convolution context. This observation motivated the development of a more localized channel-wise normalization scheme, namely, the CGN proposed in the following section.

## 3.2 | CGN

To overcome the previously identified entanglement issue, we introduce CGN, which decouples convolutional weight scaling across channels. Unlike traditional weight normalization, which learns separate gains for each feature map during training, CGN uses fixed channel-wise gains within the ODE computation and applies learnable gains only after integration. Specifically, in the ODE dynamics, we replace the trainable gain coefficients $g_k$ with a constant $1/\sqrt{c}$, where $c$ is the number of channels. This substitution results in the following simplified convolution matrix.

$$\mathbf{W}^{\mathrm{T}}\mathbf{W} = \frac{1}{c}\mathbf{V}^{\mathrm{T}}\mathbf{V}. \tag{7}$$

This fixed scaling factor removes the inter-channel gain entanglement, ensuring a more predictable and controlled gradient flow during training. Notably, the constant $1/\sqrt{c}$ binds the operator norm of $\mathbf{W}$, as formalized in the subsequent theorem. Once the ODE dynamics are integrated up to time $T$, a learnable gain matrix $\mathbf{G}_{\mathbf{CGN}}$ is applied to produce the final output $y$ as

$$y = \mathbf{G}_{\mathbf{CGN}}\mathbf{x}(T) = \begin{bmatrix} g_1^{\mathrm{CGN}} \cdot \mathbf{I} & 0 & \cdots & 0 \\ 0 & g_2^{\mathrm{CGN}} \cdot \mathbf{I} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_c^{\mathrm{CGN}} \cdot \mathbf{I} \end{bmatrix} \mathbf{x}(T), \tag{8}$$

yielding a closed-form output representation $y = e^{-\epsilon \mathrm{T}}\mathbf{G}_{\mathbf{CGN}}\left[\mathbf{x}(0) + \int_{t=0}^{\mathrm{T}} e^{\epsilon t}\left\{\mathbf{W}^{\mathrm{T}}\sigma(\mathbf{W}\mathbf{x}(t)) + \mathrm{F}(\mathbf{x}(0))\right\}dt\right]$.

This decoupled parametrization enables independent channel-wise gain tuning without interference, resulting in a more stable gradient flow within the ODE and effectively mitigating exploding gradients. Furthermore, fixing the channel-wise scaling factor to $1/\sqrt{c}$ during ODE integration allows us to bind the Lipschitz constant of the transformation formally. This approach ensures that small input perturbations are not disproportionately amplified by the ODE module, which is critical for adversarial robustness and stable training. The following theorem demonstrates this bound for $3 \times 3$ convolution filters, although extending the theorem to arbitrary $k \times k$ kernel sizes is straightforward.

**Theorem 1.** *For a $3 \times 3$ 2D convolution kernel with zero padding, where channel-wise scaling is fixed at $1/\sqrt{c}$ during ODE integration, the Lipschitz constant of the resulting convolution matrix $\mathbf{W}$ is bounded by a constant proportional to $3\sqrt{c}$. For a general $k \times k$ filtered 2D convolution with appropriate zero padding, the resulting convolution matrix has a Lipschitz constant proportional to $k\sqrt{c}$ (the following proof assumes $3 \times 3$ filters).*

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

*Proof*. Following the decomposition established by weight normalization in Section 3.1, the convolution matrix $\mathbf{W}$ can be written as $\mathbf{W} = \mathbf{G_{WN}}\mathbf{V}$, where $\mathbf{G_{WN}}$ is the gain matrix introduced by weight normalization and $\mathbf{V}$ is the gain-normalized convolutional matrix. Each block $\mathbf{V}_{j,i}$ in $\mathbf{V}$ represents the interaction between the input channel $i$ and output channel $j$, which is expressed as a block tridiagonal matrix composed of smaller tridiagonal matrices $\mathbf{B}_k^{j,i}$ for $k = 1,2,3$. Each $\mathbf{B}_k^{j,i}$ encodes $3 \times 3$ convolutional kernel coefficients $a_1^{j,i},...,a_9^{j,i}$ using the equation

$$\mathbf{B}_k^{j,i} = a_{3k-2}^{j,i}\mathbf{I}_{\downarrow} + a_{3k-1}^{j,i}\mathbf{I} + a_{3k}^{j,i}\mathbf{I}_{\uparrow}, \tag{9}$$

where $\mathbf{I}$ is the identity matrix with the upper shift matrix $\mathbf{I}_{\uparrow}$ and lower shift matrix $\mathbf{I}_{\downarrow}$. Let us define $\mathbf{P}_k$ for $k = 1,...,9$ as $\mathbf{P}_1 \triangleq \mathbf{I}_{\downarrow} \bigotimes \mathbf{I}_{\downarrow}$, $\mathbf{P}_2 \triangleq \mathbf{I}_{\downarrow} \bigotimes \mathbf{I}$, ..., $\mathbf{P}_6 \triangleq \mathbf{I} \bigotimes \mathbf{I}_{\uparrow}$, $\mathbf{P}_7 \triangleq \mathbf{I}_{\uparrow} \bigotimes \mathbf{I}_{\downarrow}$..., $\mathbf{P}_9 \triangleq \mathbf{P}_{\uparrow} \bigotimes \mathbf{P}_{\uparrow}$, where $\bigotimes$ is the Kronecker product of two matrices. Using Kronecker product notation, we have

$$\mathbf{V}_{j,i} = \mathbf{I}_{\downarrow} \bigotimes \mathbf{B}_1^{j,i} + \mathbf{I} \bigotimes \mathbf{B}_2^{j,i} + \mathbf{I}_{\uparrow} \bigotimes \mathbf{B}_3^{j,i}. \tag{10}$$

Consequently, $\mathbf{V}_{j,i}$ becomes a linear combination of Kronecker products as

$$\mathbf{V}_{j,i} = \sum_{k=1}^{9} a_k^{j,i}\mathbf{P}_k. \tag{11}$$

Aggregating these products over all channel pairs $(i,j)$ yields the following full convolutional matrix:

$$\mathbf{V} = \sum_{k=1}^{9} \left[\mathbf{a}_k^{j,i}\right] \bigotimes \mathbf{P}_k, \tag{12}$$

where $\left[\mathbf{a}_k^{j,i}\right]$ is a $c \times c$ matrix whose $(i,j)$ th element is $a_k^{j,i}$. To estimate the Lipschitz constant, we bound the operator norm $\mathbf{V}$. Using the submultiplicativity of the operator norm and inequality $\|\mathbf{A} \bigotimes \mathbf{B}\|_{\text{op}} \le \|\mathbf{A}\|_{\text{op}}\|\mathbf{B}\|_{\text{op}}$, we obtain

$$\|\mathbf{V}\|_{\text{op}} \le \sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \bigotimes \mathbf{P}_k \right\|_{\text{op}} \le \sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{op}} \cdot \|\mathbf{P}_k\|_{\text{op}}, \tag{13}$$

where the operator norm of matrix $\mathbf{A}$ is defined as

$$\|\mathbf{A}\|_{\text{op}} = \sup_{\|\mathbf{x}\|_2 = 1} \|\mathbf{A}\mathbf{x}\|_2. \tag{14}$$

Because $\|\mathbf{P}_k\|_{\text{op}} \le 1$ and $\mathbf{V}$ is a normalized convolutional matrix, we have

$$\sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{op}} \cdot \|\mathbf{P}_k\|_{\text{op}} \le \sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{op}} \le \sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{F}}, \tag{15}$$

$$\sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{op}}^2 \le \sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{F}}^2 = c. \tag{16}$$

Assuming $\left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{F}} = \sqrt{c/9}$ for all $k$, we can satisfy the above Frobenius norm condition while obtaining the upper bound on the operator norm of $\mathbf{V}$ as follows:

$$\|\mathbf{V}\|_{\text{op}} \le \sum_{k=1}^{9} \left\| \left[\mathbf{a}_k^{j,i}\right] \right\|_{\text{F}} \le \sum_{k=1}^{9} \sqrt{c/9} = \sqrt{9c}. \tag{17}$$

Finally, $\mathbf{W} = \mathbf{G_{WN}}\mathbf{V}$, and the overall Lipschitz constant becomes

$$\begin{aligned} \|\mathbf{W}\|_{op} &\le \|\mathbf{G_{WN}}\|_{\text{op}} \cdot \|\mathbf{V}\|_{\text{op}} \\ &\le \|\mathbf{G_{WN}}\|_{\text{op}} \cdot \sqrt{9c} = 3\|\mathbf{G_{WN}}\|_{\text{op}}\sqrt{c}. \end{aligned} \tag{18}$$

Therefore, the Lipschitz constant of the convolutional module is bounded by a constant determined by the kernel size and number of channels. This argument is generalized to $k \times k$ filters by defining $\mathbf{B}_k^{j,i}$ and $\mathbf{P}_k$, yielding a bound for $k\|\mathbf{G_{WN}}\|_{\text{op}}\sqrt{c}$. Because we can set $\|\mathbf{G_{WN}}\|_{\text{op}}$ to an arbitrary number using CGN, we can set the Lipschitz constant of the internal module to a constant by setting $\|\mathbf{G_{WN}}\|_{\text{op}} = \mathcal{O}\left(\frac{1}{k\sqrt{c}}\right)$, effectively prohibiting gradient explosion. □

## 4 | EXPERIMENTS

To evaluate the proposed Hopfield-type ODE module with CGN, we conducted extensive experiments across multiple datasets and backbone architectures. Our evaluation focused on three key aspects: (1) rapid robustness recovery without full model retraining, (2) comparisons with standard AT methods in terms of both robustness and clean accuracy, and (3) performance across different architectures and datasets. Note that the experimental results for the ODE module without CGN are not

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

presented here because the ODE module suffers from severe gradient explosion without CGN.

We performed evaluation on the CIFAR-10 and CIFAR-100 datasets [53] using VGG-16 with batch normalization [54], ResNet-18 [20], InceptionV3 [55], and WideResNet [56] as backbone models. The ODE module was inserted after the first activation layer in all cases. Following previous studies [19,26], the ODE damping factor $\epsilon$ was set to 0.12 across all experiments, which yields optimal performance for CIFAR-scale data. The number of ODE channels matched the feature dimension at the insertion point: 64 for VGG-16 and ResNet-18, 32 for InceptionV3, and 16 for WideResNet, thereby maintaining the topology-preserving property of neural ODEs [21].

First, we trained each backbone model on clean data without ODE modules or adversarial examples. Subsequently, three defense strategies were applied.

- *AT*: Retrains the entire model with PGD-based adversarial examples using a learning rate of 0.1.
- *Adversarial fine-tuning (AFT)*: Similar to AT but uses a lower learning rate of 0.01 to retain more original model behavior.
- *ODE module method (Ours)*: Inserts the ODE module into the pretrained model while freezing all original weights. Only ODE module parameters and batch normalization statistics are updated using Euler integration for computational efficiency.

The leakage model was updated by retraining after each leakage incident. The ODE module was inserted after the first model leakage and trained using adversarial samples generated by attacking the leaked model using a PGD attack [13]. Depending on the scenario, leakage could occur once or repeatedly to test system adaptability.

We evaluated robustness using two categories of adversarial attacks. Standard Gradient-Based Attacks *Group 1*: FGSM [12], PGD-20 [13], and CW [31]. As they were originally designed for white-box settings, we repurposed these attacks for transfer attacks by generating adversarial samples from surrogate models. Transferability-Oriented Attacks *Group 2*: MIFGSM [57], TIFGSM [34], DIFGSM [58], and NIFGSM [59]. These methods explicitly maximize adversarial transferability across models. *Group 1* evaluates general susceptibility to classical attacks in transfer settings, while *Group 2* tests performance under optimized black-box attacks. The attack parameters follow the default settings of TorchAttacks [60], except that PGD-20 uses 20 steps. The results were averaged for each group for comparison.

Figure 2 presents the training and evaluation pipelines. Figure 2A presents the AT and AFT processes,

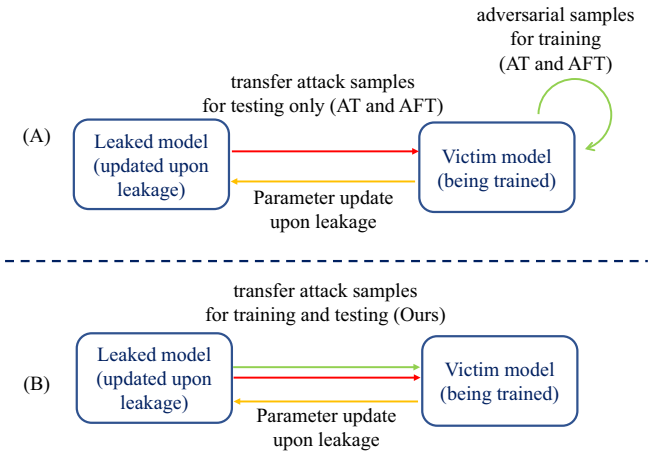


**FIGURE 2** Evaluation pipeline for different defense strategies. Red arrows represent adversarial samples used for evaluation (generated from the leaked model), while green arrows denote samples used for training. (A) For AT and AFT, adversarial training is performed using attacks generated from the current model itself. (B) The proposed method (bottom) leverages adversarial examples crafted from the leaked model, better reflecting real-world threat scenarios.

where adversarial samples are generated by attacking the current model. This approach follows conventional adversarial training protocols, assuming unrealistic attacker access to the model being updated in transfer attack scenarios.

In contrast, as shown in Figure 2B, the proposed method aligns the training process with realistic attack capabilities by using adversarial samples generated from the leaked model, which is the last model that an attacker can access, to simulate plausible transfer attacks. This distinction improves robustness under realistic threat conditions while maintaining fair comparisons. All methods were evaluated using adversarial samples from the leaked model to ensure consistent testing conditions.

## 4.1 | Single-leakage scenario

We first evaluated the robustness of the model under single-leakage scenarios, where the model parameters were exposed once at the beginning of training. The adversary retains the leaked version to generate transfer attacks during training. Each model was trained for 50 epochs, with the clean and adversarial accuracies being monitored at key intervals. The results are summarized in Table 2. Because AT consistently underperformed compared with AFT, the AT results are omitted for clarity.

Across the CIFAR-10 and CIFAR-100 datasets, our ODE module achieved a high clean accuracy and strong adversarial robustness starting in the first epoch, with

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

**TABLE 2** Adversarial accuracy comparison between AFT and the proposed ODE module.

| Dataset | Backbone | Setting | @Epoch 1 (AFT/ODE) | @Epoch 50 (AFT best/ODE) |
|---|---|---|---|---|
| CIFAR-10 | VGG16 | Clean | 0.1446/**0.8994** | 0.7691/**0.9042** |
| | | Group 1 | 0.1294/**0.8182** | 0.7560/**0.8484** |
| | | Group 2 | 0.1260/**0.7810** | 0.7455/**0.8003** |
| | InceptionV3 | Clean | 0.7132/**0.9146** | 0.8228/**0.9273** |
| | | Group 1 | 0.6987/**0.8183** | 0.8126/**0.8209** |
| | | Group 2 | 0.6857/**0.7718** | **0.8042**/0.7898 |
| CIFAR-100 | InceptionV3 | Clean | 0.4891/**0.7363** | 0.5766/**0.7447** |
| | | Group 1 | 0.4627/**0.6820** | 0.5552/**0.6820** |
| | | Group 2 | 0.4465/**0.6638** | 0.5408/**0.6638** |
| | WideResNet | Clean | 0.4974/**0.7089** | 0.6002/**0.7249** |
| | | Group 1 | 0.4653/**0.5386** | **0.5706**/0.5701 |
| | | Group 2 | **0.4463**/0.4248 | **0.5561**/0.4609 |

*Note*: Each cell shows *AFT/ODE*, where bold indicates the better value within each pair.

only 5% of the dataset. For CIFAR-10 using VGG-16, the proposed method significantly outperformed AFT in both the *Group 1* and *Group 2* adversarial scenarios. InceptionV3 on CIFAR-10 also exhibited consistent improvement, with early robustness surpassing AFT's best performance. For CIFAR-100, the proposed method remained competitive across architectures. InceptionV3 benefited the most, demonstrating strong generalization even under challenging *Group 2* transfer attacks. The performance of WideResNet is modest as a result of its shallow and wide design, which may reduce the benefits of early layer perturbation correction. The robustness against *Group 2* attacks, which are often the most challenging attack types, is well preserved, demonstrating the effectiveness of the proposed method under strong black-box transfer settings. The ODE-based defense provides immediate and sustained robustness with minimal training costs, suggesting strong applicability when full model retraining is impractical.

## 4.2 | Repeated-leakage scenario

Model leakage rarely occurs in real-world settings. In this section, we evaluate how AT, AFT, and the proposed ODE module respond when attackers periodically update their attack models using successive leakages. The initial leakage occurred before the first epoch, and the ODE module was attached to the frozen backbone model. To simulate resource constraints, only a single ODE module was inserted, and no additional modules were introduced. During training, only the ODE module and batch normalization statistics were updated, whereas the backbone remained frozen to preserve the learned knowledge. Subsequent leakages occurred every three epochs. From the second leakage onward, attackers gained access to the ODE module parameters, escalating the threat toward a white-box scenario. To stabilize training after each leakage, the ODE module was reinitialized, and a learning rate warmup [61] was applied.

Figures 3 and 4 reveal that our method recovered quickly after each model leakage, even under increasingly strong attack conditions. While all methods eventually regained accuracy, our ODE-based approach consistently outperformed the other models on both clean and adversarial samples, demonstrating rapid post-leakage adaptation. To understand this recovery behavior, we analyzed robustness during early retraining. Table 3 summarizes the adversarial accuracy of the ODE module after only 50 training batches (5% of CIFAR-10) compared with the full-epoch and peak AFT performance. The ODE module not only surpasses AFT's best performance with a fraction of the data but also exhibits low variance, indicating stable and reliable adaptation. These results support a broader defense paradigm, namely, *adversarial defense by computation*. Even white-box attacks that exploit updated gradients can be effectively mitigated if the ODE module is reinitialized rapidly enough to preempt the attacker's ability to leverage newly exposed parameters. Although Table X reveals that the ODE module approaches AFT's peak robustness with substantially fewer updates, Figure 5 further reveals that this rapid adaptation behavior persists even under limited data settings.

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

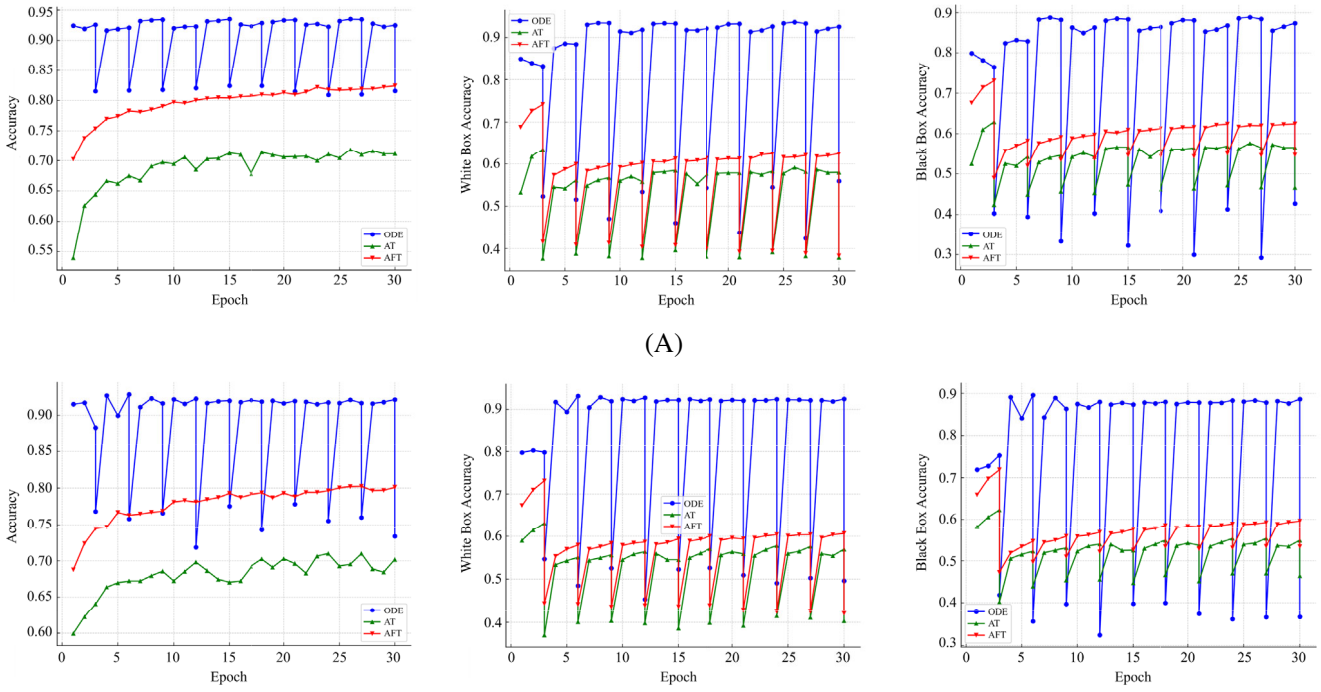


**FIGURE 3** Robustness under repeated model leakage on CIFAR-10. (A) InceptionV3 and (B) ResNet-18 backbone results. Columns (left to right): clean accuracy, *Group 1* (white-box) attack accuracy, and *Group 2* (black-box) attack accuracy. Our ODE module exhibits rapid robustness recovery and higher performance after each model leakage compared with the baseline defenses.

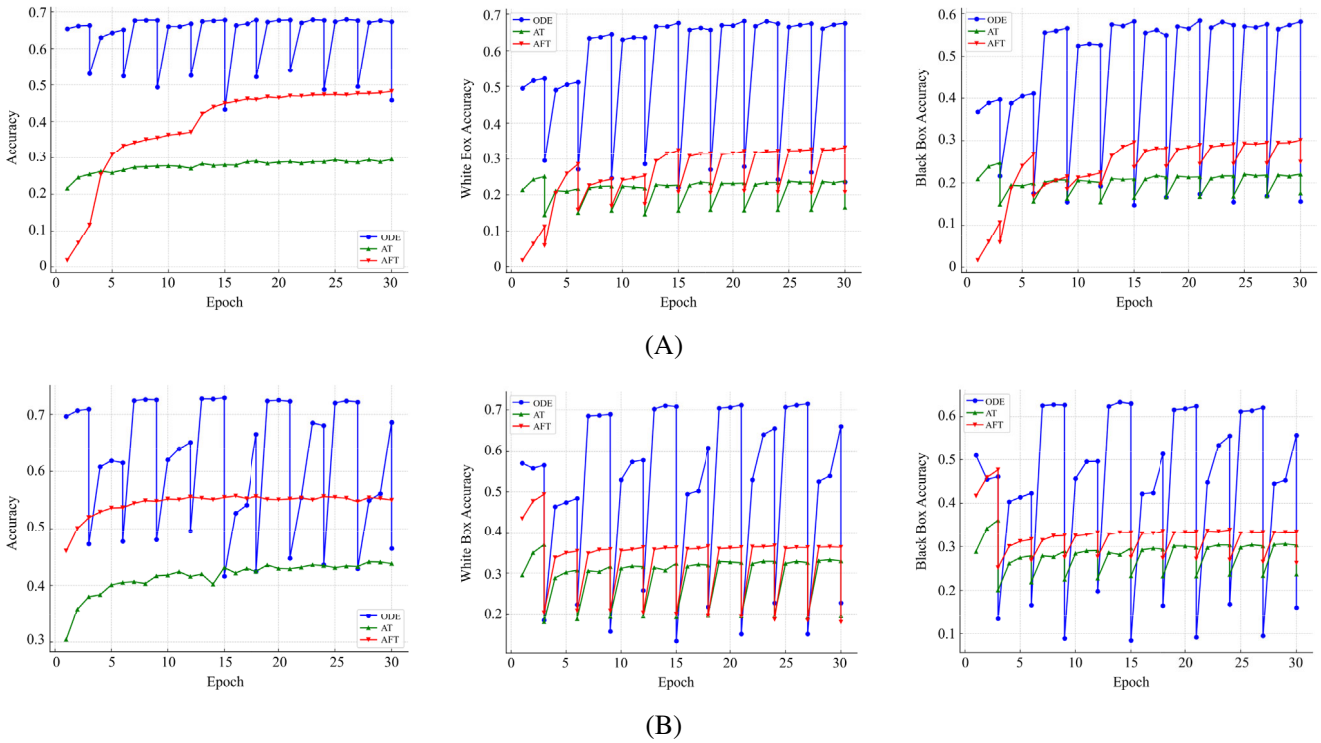


**FIGURE 4** Robustness under repeated model leakage on CIFAR-100. (A) VGG-16 and (B) ResNet-18 backbone results. Left to right: clean accuracy, *Group 1* (white-box) attacks, and *Group 2* (black-box) attacks. Our ODE module exhibits rapid recovery and maintains robustness even after repeated leakages, outperforming the baseline defenses.

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

**TABLE 3** Comparison of adversarial accuracy: ODE after 50 batches (=5% of the training data) vs. AFT in the first epoch and AFT's best performance on CIFAR-10.

| Backbone | Method | Group 1 | Group 2 |
|---|---|---|---|
| WideResNet | AFT (1 epoch) | 0.4219 | 0.5133 |
| | ODE (50 batches) | 0.7470 | 0.7319 |
| | AFT Best (30 ep.) | 0.7497 | 0.7344 |
| InceptionV3 | AFT (1 epoch) | 0.6880 | 0.6761 |
| | ODE (50 batches) | 0.7416 | 0.7310 |
| | AFT Best (30 ep.) | 0.8163 | 0.7655 |

**TABLE 4** Accuracy for different ODE module insertion depths at Epoch 10 on the CIFAR-100 dataset. For each backbone model, the better performance is shown in bold for each input type.

| Backbone | Depth | Clean | Group 1 | Group 2 |
|---|---|---|---|---|
| VGG16 | 2 | **0.5737** | **0.4564** | **0.4004** |
| | 4 | 0.2119 | 0.2300 | 0.3007 |
| ResNet18 | 2 | **0.5978** | **0.5819** | **0.5332** |
| | 4 | 0.4898 | 0.3491 | 0.3138 |

*Note*: Shallow placement (Depth 2) consistently outperforms deeper placement (Depth 4) across clean and adversarial inputs.

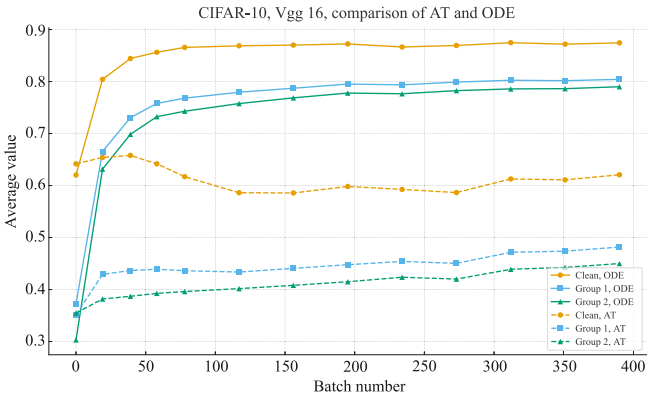


**FIGURE 5** Comparison of data efficiency between AT (dotted line) and the proposed ODE method (solid line) using the VGG-16 model on CIFAR-10. Each curve represents the average robustness recovery over 10 post-breach training epochs. ODE achieves faster and more stable robustness recovery with only a small fraction of training data compared with standard AT.

## 4.3 | Impact of ODE module placement depth

We hypothesize that adversarial perturbations are amplified as they propagate through the deeper layers of a neural network. To intercept such distortions early, we inserted the ODE module after the first activation layer in the previous experiments. To test the effect of ODE module depth, we conducted experiments in which the ODE module was placed after the second and fourth activation layers. Table 4 compares the performances at two insertion depths (2 and 4) on one CIFAR-100. Consistently, a shallow placement (Depth 2) yielded stronger clean and adversarial accuracy across all settings. This trend supports our design choice, indicating that the ODE module is most effective as an early corrective mechanism, steering adversarial noise away from sensitive feature directions without degrading clean performance. This result is consistent with the experimental results of the WideResNet model, where the ODE module yielded a relatively small performance gain. Because WideResNet trades depth for width, a single WideResNet layer can affect a transformation comparable to that of several layers in a deep, narrow network. Consequently, the ODE module inserted into WideResNet was effectively placed at a greater depth.

## 4.4 | Training efficiency and runtime analysis

To complement our data efficiency analysis, we also examined the actual wall-clock training time of the proposed ODE module compared with AT and AFT. Although not all experiments were performed on identical hardware, we conducted controlled measurements for two representative configurations in the same GPU environment (NVIDIA A100).

For the SVHN dataset with the WideResNet backbone, one epoch of adversarial training using our ODE module required approximately 871 s to 872 s, compared with 920 to 929 s for AT and AFT, corresponding to an approximately 6% to 7% reduction in per-epoch training time. Similarly, on the CIFAR-10 dataset with the InceptionV3 backbone, the ODE-based method completed one epoch in approximately 699 to 700 s, whereas AT/AFT required at least 717 s (approximately 2% to 3% training time reduction). These results indicate that the proposed approach achieves comparable or slightly better computational efficiency while maintaining its key advantage of requiring only 5% of the training data to recover adversarial robustness following model leakage.

Overall, even when the wall-clock difference is marginal, the ODE module's efficiency becomes more significant in practice because it converges rapidly with minimal data, enabling quick post-leakage adaptation without sacrificing clean accuracy.

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

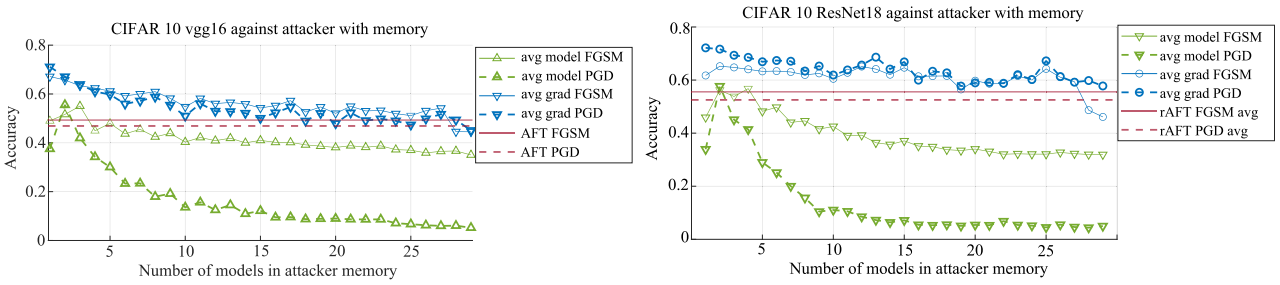


**FIGURE 6** Robustness against memory-aware attacks on the CIFAR-10 dataset. Classification accuracy under *Group 1* and *Group 2* attacks, where the attacker stores and aggregates information from up to 30 previously leaked models. The "avg grad" strategy maintains stable robustness, whereas the "avg model" and AFT deteriorate as the attacker's memory grows.

### 4.5 | Memory-aware attacks

Our previous evaluations assumed that an attacker can access only the latest leaked model. However, real-world adversaries may store and exploit multiple leaks over time, aggregating knowledge in ways resembling white-box access. Athalye and others [47] demonstrated that gradient obfuscation fails when attackers average over multiple views. We simulated this advanced threat by assuming that an attacker stores up to 30 leaked models and generates adversarial examples using either averaged weights (*avg model*) or averaged gradients (*avg grad*). As the number of stored models grows, these attacks become more powerful.

To address this type of attack, we designed memory-aware defense strategies using progressively richer attacker models during training.

- *AFT FGSM/PGD avg*: AFT using weak attacks (gradients from a single leaked model).
- *avg model*: ODE module trained using adversarial examples from averaged weights of previously leaked models.
- *avg grad*: ODE module trained using adversarial examples from averaged gradients across all stored models.

Figure 6 illustrates how the classification accuracy changes across the datasets and backbones as the number of stored leaked models increases. Among the evaluated strategies for combating memory-aware attacks, the performance of the "avg model" degraded significantly, with performance dropping sharply after only 10 to 12 leakages. In contrast, the "avg grad" method, where the ODE module is trained using averaged gradients from all leaked models, maintains stable robustness up to 30 leaks, even under strong *Group 2* attacks, representing a significant improvement over the NEO model presented by Shan and others [50], which endured at most 8 to 10 leakages.

These findings indicate that storing the gradient information of previously leaked models to handle memory-equipped attacks can significantly delay robustness degradation. By training on evolving attack signals, our method effectively prolongs defensive reliability under persistent leakage scenarios.

## 5 | LIMITATIONS AND FUTURE WORK

Although our work demonstrates the feasibility of rapid robustness recovery on CIFAR-scale datasets, several directions warrant further investigation. A deeper understanding of internal dynamics, such as how different attack types interact with ODE flows and how feature trajectories evolve across network depths, would provide valuable insights for developing more principled defense frameworks. Additionally, extending the proposed approach to large-scale, high-resolution datasets and exploring automated architecture-aware module designs could broaden its applicability to diverse real-world scenarios, including vision transformers and foundation models, where full retraining is prohibitively expensive.

## 6 | CONCLUSION

We proposed a plug-in defense module based on Hopfield-type neural ODEs to counter transfer attacks under model leakage scenarios. Our key innovation is CGN, which analytically bounds the Lipschitz constant to ensure stable ODE dynamics. Unlike conventional adversarial training, which requires full model retraining, our approach enables rapid robustness recovery using only 5% of the training data while freezing the backbone weights. The proposed module effectively handles single- and repeated-leakage scenarios using reinitialization and learning rate warmup strategies. Extensive experiments

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

on CIFAR-10 and CIFAR-100 demonstrated consistent improvements over adversarial training in terms of both clean and adversarial accuracy, maintaining robustness even against memory-aware attackers storing multiple leaked models. These results establish a new defense paradigm, which we refer to as *adversarial defense by computation*, where rapid adaptation outpaces attacker exploitation of leaked parameters, transforms white-box attacks into manageable transfer scenarios, and offers practical solutions when full retraining is impractical.

## CONFLICT OF INTEREST STATEMENT

The authors declare that there are no conflicts of interest.

## ORCID

*Kimin Yun* https://orcid.org/0000-0002-4493-9437

## REFERENCES

1. S. Baek and J. T. Lee, *High-speed and precise virtual try-on with two-stage semantic segmentation and a latent consistency model for optimized diffusion processes*, ETRI J. **47** (2025), no. 5, 881–892. DOI 10.4218/etrij.2024-0592
2. K. Yun, H.-I. Kim, K. Bae, and J. Moon, *Background memory-assisted zero-shot video object segmentation for unmanned aerial and ground vehicles*, ETRI J. **45** (2023), no. 5, 795–810. DOI 10.4218/etrij.2023-0115
3. K. Yun, Y. Kwon, S. Oh, J. Moon, and J. Park, *Vision-based garbage dumping action detection for real-world surveillance platform*, ETRI J. **41** (2019), no. 4, 494–505. DOI 10.4218/etrij.2018-0520
4. P. Gao, L. Xu, W.-J. Tang, F. Wang, H. Fujita, H. Aljuaid, and R.-Y. Yuan, *Towards patch-based noise compression for adversarial attack against transformer-based visual tracking*, IEEE Trans. Inf. Forensics Secur. **21** (2025), 854–869.
5. 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 **23** (2021), no. 10, 1359.
6. L. Meunier, J. Atif, and O. Teytaud, *Yet another but more efficient black-box adversarial attack: tiling and evolution strategies*, arXiv preprint (2019). DOI 10.48550/arXiv.1910.02244
7. N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, *Practical black-box attacks against deep learning systems using adversarial examples*, arXiv preprint (2016). DOI 10.48550/arXiv.1602.02697
8. R. Bommasani, D. A. Hudson, E. Adeli, R. Altman, S. Arora, S. von Arx, et al., *On the opportunities and risks of foundation models*, arXiv preprint (2021). DOI 10.48550/arXiv.2108.07258
9. C. Sitawarin, J. Chang, D. Huang, W. Altoyan, and D. Wagner, *Defending against transfer attacks from public models*, 2023. arXiv preprint (2024). DOI 10.48550/arXiv.2310.17645
10. J. Zhang, J. Sang, Q. Yi, and C. Xu, *Introducing foundation models as surrogate models: advancing towards more practical adversarial attacks*, arXiv preprint (2018).
11. H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El haoui, and M. Jordan, *Theoretically principled trade-off between robustness and accuracy*, (Int. Conf. Mach. Learn., PMLR, Long Beach, CA Montréal, Canada, USA), 2019, pp. 7472–7482.
12. I. J. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, arXiv preprint (2014). DOI 10.48550/arXiv.1412.6572
13. A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, *Towards deep learning models resistant to adversarial attacks*, arXiv preprint (2017). DOI 10.48550/arXiv.1706.06083
14. C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, *Countering adversarial images using input transformations*, arXiv preprint (2017). DOI 10.48550/arXiv.1711.00117
15. Y. Bai, Y. Zeng, Y. Jiang, S.-T. Xia, X. Ma, and Y. Wang, *Improving adversarial robustness via channel-wise activation suppressing*, arXiv preprint (2021). DOI 10.48550/arXiv.2103.08307
16. W. J. Kim, Y. Cho, J. Jung, and S.-E. Yoon, Feature separation and recalibration for adversarial robustness, (Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit., IEEE, Vancouver, BC, Canada), 2023, pp. 8183–8192.
17. E. Demirci, F. Karakoç, and A. Kütahyalıoglu, *Mittag-leffler stability of neural networks with Caputo–Hadamard fractional derivative*, Math. Method. Appl. Sci. **47** (2024), no. 12, 10091–10100.
18. R. T. Q. Chen, Y. Rubanova, J. Bettencourt, and D. Duvenaud, Neural ordinary differential equations, (Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montreal, Canada) 2018, pp. 6572–6583.
19. Y.-H. Shin and S. J. Baek, *Hopfield-type neural ordinary differential equation for robust machine learning*, Pattern Recog. Lett. **152** (2021), 180–187.
20. K. He, X. Zhang, S. Ren, and J. Sun, *Deep residual learning for image recognition*, (Proc. IEEE Conf. Comput. Vision Pattern Recognit., Las Vegas, NV, USA) 2016, pp. 770–778.
21. E. Dupont, A. Doucet, and Y. W. Teh, Augmented neural ODEs, (Conference on Neural Information Processing Systems, Vancouver, BC, Canada), 2019, pp. 3140–3150.
22. F. Carrara, R. Caldelli, F. Falchi, and G. Amato, *On the robustness to adversarial examples of neural ode image classifiers*, (2019 IEEE Int. Workshop Inf. Forensics Secur. (WIFS), Delft, The Netherlands), 2019, pp. 1–6.
23. H. Yan, J. Du, V. incentY. F. Tan, and J. Feng, *On robustness of neural ordinary differential equations*, arXiv preprint (2019). DOI 10.48550/arXiv.1910.05513
24. X. Liu, T. Xiao, S. Si, Q. Cao, S. Kumar, and C.-J. Hsieh, *Neural SDE: stabilizing neural ODE networks with stochastic noise*, 2019. arXiv preprint (2019). DOI 10.48550/arXiv:1906.02355
25. Q. Kang, Y. Song, Q. Ding, and W. P. Tay, Stable neural ODE with Lyapunov-stable equilibrium points for defending against adversarial attacks, (Proceedings of the 35th International Conference on Neural Information Processing Systems), 2021, pp. 14925–14937.
26. Z. Yang, Y. Liu, C. Bao, and Z. Shi, Interpolation between residual and non-residual networks, (Int. Conf. Mach. Learn., Vienna, Austria), 2020, pp. 10736–10745.
27. V. Purohit, *Ortho-ODE: enhancing robustness and of neural ODEs against adversarial attacks*, arXiv preprint (2023), DOI 10.48550/arXiv:2305.09179
28. H. Luo, T. He, and Z. Yi, *A stable mapping of nmODE*, Artif. Intell. Rev. **57** (2024), no. 5, 120.
29. Q. Yang, X. Hu, J. Yu, Q. Sun, L. Shu, Z. Yi, and Y. Liao, *var-nmODE: model with L2-stability based on nmODE for defending against adversarial attacks*, Neurocomputing **2025** (2025), 130605.

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

30. D. Zeng, Z. Zuo, L. Yang, X. Xiao, and Z. Tang, *Text adversarial attacks using policy gradients against deep learning classifiers*, ETRI J. **47** (2025), no. 6, 1085–1103. DOI 10.4218/etrij.2024-0339
31. N. Carlini and D. Wagner, *Towards evaluating the robustness of neural networks*, (2017 IEEE Symp. Secur. Privacy (SP), San Jose, CA, USA), 2017, pp. 39–57.
32. J. Zhang, W. Wu, J. Huang, Y. Huang, W. Wang, Y. Su, and M. R. Lyu, *Improving adversarial transferability via neuron attribution-based attacks*, (Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., New Orleans, LA, USA), 2022, pp. 14993–15002.
33. N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, *Practical black-box attacks against machine learning*, (Proc. 2017 ACM Asia Conf. Comput. Commun. Secur.,Abu Dhabi, United Arab Emirates), 2017, pp. 506–519.
34. Y. Dong, T. Pang, H. Su, and J. Zhu, *Evading defenses to transferable adversarial examples by translation-invariant attacks*, (Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA), 2019, pp. 4312–4321.
35. Y. Liu, X. Chen, C. Liu, and D. Song, *Delving into transferable adversarial examples and black-box attacks*, arXiv preprint (2016). DOI 10.48550/arXiv:1611.02770
36. R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, *Learning with a strong adversary*, arXiv preprint (2015). DOI 10.48550/arXiv:1511.03034
37. M. Carletti, A. Sinigaglia, M. Terzi, and G. A. Susto, *On the limitations of adversarial training for robust image classification with convolutional neural networks*, Inform. Sci. **675** (2024), 120703.
38. Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, *Improving adversarial robustness requires revisiting misclassified examples*, (Int. Conf. Learn. Representations, OpenReview, New Orleans, Louisiana, USA), 2020.
39. M. Dong, Y. Li, Y. Wang, and C. Xu, *Adversarially robust neural architectures*, IEEE Trans. Pattern Anal. Mach. Intell. **47** (2025), 4183–4197.
40. P. Pauli, A. Koch, J. Berberich, P. Kohler, and F. Allgöwer, *Training robust neural networks using Lipschitz bounds*, IEEE Control Syst. Lett. **6** (2021), 121–126.
41. A. Virmaux and K. Scaman, Lipschitz regularity of deep neural networks: analysis and efficient estimation, (Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, Canada), 2018, pp. 3835–3844.
42. A. Araujo, B. Negrevergne, Y. Chevaleyre, and J. Atif, *On Lipschitz regularization of convolutional layers using Toeplitz matrix theory*, (Proc. AAAI Conf. Artif. Intell.), 2021, pp. 6661–6669.
43. H. Sedghi, V. Gupta, and P. M. Long, *The singular values of convolutional layers*, arXiv preprint (2018). DOI 10.48550/arXiv:1805.10408
44. P. Mangla, V. Singh, S. Havaldar, and V. Balasubramanian, *On the benefits of defining vicinal distributions in latent space*, Pattern Recognit. Lett. **152** (2021), 382–390.
45. J. Alagurajah and C.-H. H. Chu, *Adversarial defense by restricting in-variance and co-variance of representations*, (Proc. 37th ACM/SIGAPP Symp. Appl. Comput.), 2022, pp. 58–65.
46. F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, *Defense against adversarial attacks using high-level representation guided denoiser*, (Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA), 2018, pp. 1778–1787.
47. A. Athalye, N. Carlini, and D. Wagner, *Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples*, (Int. Conf. Mach. Learn., Stockholm, Sweden), 2018, pp. 274–283.
48. D. Madaan, J. Shin, and S. J. Hwang, *Adversarial neural pruning with latent vulnerability suppression*, (Int. Conf. Mach. Learn., Vienna, Austria), 2020, pp. 6575–6585.
49. C. Zhang, A. Liu, X. Liu, Y. Xu, H. Yu, Y. Ma, and T. Li, *Interpreting and improving adversarial robustness of deep neural networks with neuron sensitivity*, IEEE Trans. Image Process. **30** (2020), 1291–1304.
50. S. Shan, W. Ding, E. Wenger, H. Zheng, and B. Y. Zhao, *Post-breach recovery: protection against white-box adversarial examples for leaked DNN models*, (Proc. 2022 ACM SIGSAC Conf. Comput. Commun. Secur., Los Angeles, CA, USA), 2022, pp. 2611–2625.
51. Y. Bengio, P. Simard, and P. Frasconi, *Learning long-term dependencies with gradient descent is difficult*, IEEE Trans. Neural Netwo. **5** (1994), no. 2, 157–166.
52. T. Salimans and D. P. Kingma, Weight normalization: a simple reparameterization to accelerate training of deep neural networks, (Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain), 2016, pp. 901–909.
53. A. Krizhevsky and G. Hinton, *Learning multiple layers of features from tiny images*, Technical Report, University of Toronto, 2009.
54. K. Simonyan and A. Zisserman, *Very deep convolutional networks for large-scale image recognition*, arXiv preprint (2014). DOI 10.48550/arXiv:1409.1556
55. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, *Rethinking the inception architecture for computer vision*, (Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Las Vegas, NV, USA), 2016, pp. 2818–2826.
56. S. Zagoruyko and N. Komodakis, *Wide residual networks*, arXiv preprint (2016). DOI 10.48550/arXiv:1605.07146
57. Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, *Boosting adversarial attacks with momentum*, (Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA), 2018, pp. 9185–9193.
58. C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, *Improving transferability of adversarial examples with input diversity*, (Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA), 2019, pp. 2730–2739.
59. J. Lin, C. Song, K. He, L. Wang, and J. E. Hopcroft, *Nesterov accelerated gradient and scale invariance for adversarial attacks*, arXiv preprint (2019). DOI 10.48550/arXiv:1908.06281
60. H. Kim, *Torchattacks: a PyTorch repository for adversarial attacks*, arXiv preprint (2020). DOI 10.48550/arXiv:2010.01950
61. T. He, Z. Zhang, H. Zhang, Z. Zhang, J. Xie, and M. Li, *Bag of tricks for image classification with convolutional neural networks*, (Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Long Beach, CA, USA), 2019, pp. 558–567.


22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

## AUTHOR BIOGRAPHIES

**Yu-Hyun Shin** received his B.S. degree and joint M.S./Ph.D. degree in Computer Science from Korea University, Republic of Korea, in 2014 and 2022, respectively. He is currently a senior researcher at the Automotive Driving Technology Research Division of the Korea Automotive Technology Institute (KATECH) in Cheonan, Republic of Korea. His research interests include sensor fusion, domain adaptation, transfer learning, long-tailed learning, and multimodal learning.

**Kimin Yun** received his B.S. degree in Electrical Engineering from Seoul National University, Republic of Korea, in 2010. He then received his Ph.D. in Electrical and Computer Engineering from Seoul National University, Republic of Korea, in 2017, through an integrated M.S./Ph.D. program. Since 2017, he has worked with the Visual Intelligence Research Section of the Artificial Intelligence Research Laboratory at the Electronics and Telecommunications Research Institute (ETRI) in Daejeon, Republic of Korea. Additionally, in 2025, he joined the University of Science and Technology (UST) as an associate professor of Artificial Intelligence. His current research interests include machine learning and computer vision, with a focus on real-world challenges and the development of practical applications.

**Yuseok Bae** received his B.S. and M.S. degrees in Computer Science from Kyungpook National University, Republic of Korea, in 1995 and 1997, respectively. He is currently a principal researcher in the Visual Intelligence Research Section of the Electronics and Telecommunications Research Institute (ETRI), where he has been involved in various projects such as distributed computing, home network middleware, IPTV middleware, big data analytics, and visual artificial intelligence since 1997. He received his Ph.D. in Computer Science from Kyungpook National University in 2011. His research interests include computer vision, machine learning, deep learning, and artificial intelligence.

**How to cite this article:** Y.-H. Shin, K. Yun, and Y. Bae, *Fast adversarial robustness recovery from model parameter exposure through Hopfield-type neural ordinary differential equations*, ETRI Journal (2026), 1–15, DOI 10.4218/etrij.2025-0264

22337326, 0, Downloaded from https://onlinelibrary.wiley.com/doi/10.4218/etrij.2025-0264 by Electronics and Telecommunications, Wiley Online Library on [28/06/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License