

FastJSMA: Accelerating Jacobian-based Saliency Map Attacks through Gradient Decoupling

Zhenghao Gao^{1,*} Shengjie Xu^{2,*} Zijing Li² Meixi Chen³ Chaojian Yu⁴ Yuanjie Shao⁴ Changxin Gao^{1,†}

¹School of Artificial Intelligence and Automation, Huazhong University of Science and Technology

²School of Software Engineering, Huazhong University of Science and Technology

³Journalism and Information Communication School, Huazhong University of Science and Technology

⁴School of Electronic Information and Communications, Huazhong University of Science and Technology

{u202215226, u202317280, u202317273, u202317218, yuchaojian, shaoyuanjie, cgao}@hust.edu.cn

Abstract

Adversarial attack plays a critical role in evaluating the robustness of deep learning models. Jacobian-based Saliency Map Attack (JSMA) is an interpretable adversarial method that offers excellent pixel-level control and provides valuable insights into model vulnerabilities. However, its quadratic computational complexity $O(M^2 \times N)$ renders it impractical for large-scale datasets, limiting its application despite its inherent value. This paper proposes FastJSMA, an efficient attack method that addresses these computational limitations. Our approach introduces a gradient decoupling mechanism that decomposes the Jacobian calculation into complementary class suppression (g^-) and class excitation (g^+) gradients, reducing complexity to $O(M\sqrt{N})$. Additionally, we implement a class probing mechanism and an adaptive saliency threshold to further optimize the process. Experimental results across multiple datasets demonstrate that FastJSMA maintains comparable attack success rates while dramatically reducing computation time—requiring only 2.9% of JSMA’s processing time on CIFAR-10 and 1.2% on CIFAR-100, and successfully operating on ImageNet where traditional JSMA fails due to memory constraints. This advancement enables the practical application of interpretable saliency map-based attacks on large-scale datasets, balancing effectiveness with computational efficiency.

1. Introduction

1.1. Research Background

Deep learning has achieved breakthrough progress in computer vision, natural language processing, and other fields,

but its security issues have increasingly attracted attention. In 2013, Szegedy et al. [1] first discovered the sensitivity of deep neural networks to small perturbations, i.e., by adding imperceptible perturbations to inputs, models can be induced to produce incorrect predictions. This discovery revealed potential vulnerabilities in deep learning systems and drove rapid development in adversarial attack research.

Adversarial attack techniques have evolved from simple approaches like the Fast Gradient Sign Method (FGSM) [2] to complex methods such as PGD [3] and C&W attacks [4], each offering different trade-offs between effectiveness, efficiency, and perturbation magnitude.

1.2. Research Motivation

The Jacobian-based Saliency Map Attack (JSMA) [5] is an adversarial method that identifies and modifies the most influential pixels through Jacobian matrix analysis. It offers unique advantages including intuitive visualization through saliency maps that enhance interpretability, precise control over modified pixels, and strong effectiveness across various datasets. Despite these merits and attempts to improve it through weighted approaches like WJSMA [6], JSMA remains largely underutilized due to prohibitive computational costs. Its full Jacobian matrix calculation requires $O(M^2 \times N)$ complexity (where M is the number of pixels and N is the number of classes), creating an insurmountable barrier for large-scale applications like ImageNet [7].

This limitation stems from three factors: (a) separate backward propagation for each output node; (b) saliency evaluations across the entire Jacobian matrix; and (c) repeated calculations across iterations. Consequently, the adversarial machine learning community has effectively abandoned JSMA’s valuable properties due to computational constraints rather than methodological limitations.

^{0*}Equal contribution

^{0†}Corresponding author

1.3. Contributions

To revitalize this valuable but computationally constrained method, we propose FastJSMA (Fast Jacobian-based Saliency Map Attack), an efficient adversarial attack based on gradient decoupling approximation:

- **Gradient Decoupling Mechanism:** We decompose the Jacobian matrix into complementary class suppression (g^-) and excitation (g^+) gradients, reducing computational complexity from $O(M^2 \times N)$ to $O(M\sqrt{N})$ while maintaining attack effectiveness through dynamic weight allocation.
- **Class Probing and Adaptive Threshold:** We introduce a square-root sampling heuristic to efficiently identify vulnerable decision boundaries and an adaptive threshold mechanism to prioritize influential pixels, optimizing both efficiency and effectiveness.
- **Comprehensive Evaluation:** Our experiments demonstrate dramatic efficiency gains—requiring only 2.9% of JSMA’s processing time on CIFAR-10 and achieving up to 81.4× speedup on CIFAR-100 while maintaining comparable attack success rates. Most notably, FastJSMA successfully processes ImageNet samples, where traditional JSMA and WJSMA fail entirely due to memory constraints.

2. Related Work

2.1. Overview of Adversarial Attack Methods

Adversarial attacks can be categorized as white-box or black-box attacks based on the attacker’s knowledge of the model. White-box attacks assume the attacker has complete access to the model structure and parameters, with representative methods including FGSM [2] and PGD [3]. Black-box attacks can only access the model’s input-output interface, primarily relying on transferability [8] or query optimization [9]. Recent work has also focused on improving adversarial training defenses [10–12].

From the perspective of attack objectives, they can be divided into targeted and non-targeted attacks. Targeted attacks aim to misclassify input samples as specified target classes, such as DeepFool [13]; non-targeted attacks only need to cause incorrect predictions, regardless of the specific error class, such as FGSM. These two types of attack methods have their respective advantages in practical applications: targeted attacks enable more precise control, while non-targeted attacks typically have higher success rates and lower computational costs.

Gradient-based adversarial attacks have evolved significantly. After FGSM was proposed, researchers developed improved variants: (1) MI-FGSM [14] introduces momentum terms; (2) NI-FGSM [15] utilizes Nesterov accelerated gradients; (3) TI-FGSM [16] enhances transferability through translation invariance; (4) DI-FGSM [17] adopts

diversified input strategies. Recent research by Mao et al. [18] discovered that adversarial attacks are reversible with natural supervision, providing new insights into the fundamental properties of adversarial perturbations. These methods optimize gradient-based attack strategies from different perspectives but have not fully addressed the issue of interpretability.

2.2. Detailed Analysis of JSMA

As an interpretable attack method, JSMA was proposed by Papernot et al. [5], with its core idea being the use of the Jacobian matrix to identify input features that have the greatest impact on model output. This approach builds upon fundamental research in saliency detection, such as the boolean map approach [19], which provides theoretical foundations for identifying visually significant regions. Specifically, for input x and target class t , JSMA first calculates the Jacobian matrix:

$$J_F(x) = \frac{\partial F(x)}{\partial x} = \left[\frac{\partial F_j(x)}{\partial x_i} \right]_{i,j} \quad (1)$$

where $F_j(x)$ represents the model’s prediction probability for class j . Based on the Jacobian matrix, JSMA defines the saliency map S :

$$S(x, t)[i] = \begin{cases} 0, & \text{if } \frac{\partial F_t(x)}{\partial x_i} < 0 \text{ or } \sum_{j \neq t} \frac{\partial F_j(x)}{\partial x_i} > 0 \\ \left| \frac{\partial F_t(x)}{\partial x_i} \right| \cdot \left| \sum_{j \neq t} \frac{\partial F_j(x)}{\partial x_i} \right|, & \text{otherwise} \end{cases} \quad (2)$$

Subsequently, researchers have made various improvements to JSMA: (1) JSMA++ [8] enhances attack efficiency by optimizing saliency calculations; (2) One-pixel JSMA [20] restricts modifications to a single pixel, achieving more precise control; (3) Adaptive JSMA [21] introduces adaptive threshold strategies, improving attack success rates. The concept of saliency maps has also been extended to other domains, such as 3D point clouds [22], demonstrating the versatility of saliency-based approaches across different data modalities.

An important advancement is the Weighted JSMA (WJSMA) [6], which addresses limitations in the original JSMA’s saliency calculation. WJSMA penalizes gradients associated with small probabilities to create more balanced saliency maps. WJSMA incorporates class probabilities as weights in the saliency calculation, setting the saliency to zero when specific gradient conditions are not met. This weighting mechanism makes WJSMA faster than the original JSMA while maintaining good balance between attack success rate and computational cost. However, even with these improvements, WJSMA still inherits the fundamental computational inefficiency of JSMA for large-scale datasets, as it retains the $O(M^2 \times N)$ complexity that creates memory and processing bottlenecks on datasets like ImageNet.

2.3. Related Optimization Techniques

To address the efficiency problems of adversarial attacks in large-scale scenarios, researchers have proposed various optimization strategies. One class of methods improves efficiency by enhancing gradient computation, such as One-Pixel Attack [20], which reduces search space through evolutionary algorithms, and SparseFool [23], which optimizes computational efficiency by leveraging sparsity.

Another class of methods proposes specialized optimization strategies for large-scale datasets. For example, Fast Gradient Method [24] enhances attack efficiency through ensemble techniques, while AutoAttack [25] adopts automated strategies to select optimal attack parameters.

Recent research has also explored algorithmic optimizations specifically targeting the computational complexity of gradient-based attacks. These approaches include gradient approximation techniques [26], adaptive norm constraints for efficient minimum-norm attacks [27], and decomposition strategies that reduce the dimensionality of the optimization space [21].

These optimization methods have their respective advantages and disadvantages: gradient-based methods offer high computational efficiency but lower interpretability; optimization-based methods provide good results but incur high computational costs; heuristic-based methods are simple and fast but have unstable success rates. In comparison, our proposed FastJSMA maintains JSMA’s interpretability while significantly improving computational efficiency through dual-gradient approximation, effectively reducing the complexity from $O(M^2 \times N)$ to $O(M\sqrt{N})$ and making saliency map-based attacks practical for large-scale datasets for the first time.

3. Methodology

This section details the technical principles of FastJSMA. We first establish the mathematical model of the problem, then elaborate on the design rationale of the dual-gradient approximation strategy, followed by a detailed explanation of the saliency map generation process, and finally theoretically analyze the computational complexity of the algorithm.

3.1. Problem Modeling and Attack Objectives

For an input sample $x \in \mathbb{R}^{C \times H \times W}$ and a target classification model $f : \mathbb{R}^{C \times H \times W} \rightarrow \mathbb{R}^N$, where C, H, W represent the number of channels, height, and width respectively, and N is the total number of classes. Given the original predicted class $t = \arg \max f(x)$, the goal of the adversarial attack is to find a perturbation δ such that:

$$\arg \max f(x + \delta) \neq t \quad \text{s.t.} \quad \|\delta\|_0 \leq \epsilon \quad (3)$$

where ϵ represents the allowed number of pixels to modify, and $\|\cdot\|_0$ is the L_0 norm. The reasons for choosing the L_0 norm constraint rather than other norms (such as L_2 or L_∞) are: (1) L_0 norm directly limits the number of modified pixels, conforming to the principle of minimal intervention; (2) a limited number of modifications makes the attack process more interpretable and understandable; (3) maintains consistent constraint forms with traditional JSMA for fair comparison.

Traditional JSMA computes the complete Jacobian matrix $J = \nabla_x f(x)$ with $O(M^2 \times N)$ complexity, where M represents the total number of pixels and N denotes the number of classification classes. This substantial computational burden and memory requirements render JSMA impractical for large-scale datasets and real-world applications.

FastJSMA addresses this limitation through a dual-gradient approximation strategy that reduces complexity to $O(M\sqrt{N})$ while maintaining attack effectiveness.

3.2. Target Function Construction

The core mathematical insight that enables FastJSMA’s efficiency breakthrough lies in our recognition that the Jacobian-based saliency computation in traditional JSMA can be decomposed into two complementary objectives: first, decreasing the probability of the current class, and second, increasing the probability of alternative classes. This decomposition forms the theoretical foundation of our dual-gradient approximation strategy.

Analyzing the mathematical structure of traditional JSMA reveals that its saliency map computation essentially measures two gradient components for each pixel: how modifying that pixel decreases the current class score and how it increases other class scores. While traditional JSMA explicitly computes these gradients through the full Jacobian matrix, we observed that these two objectives could be approximated through carefully designed loss functions without requiring the prohibitively expensive Jacobian computation.

To avoid computing the complete Jacobian matrix, we propose a dual-gradient approximation strategy. The core idea of this strategy is to construct two complementary objective functions that simulate the enhancement effect on the target class and the suppression effect on other classes in JSMA. The specific definitions are as follows:

1. **Class Suppression Gradient:** Construct a virtual label $\hat{y}_i^- \in \mathbb{R}^N$, where

$$\hat{y}_i^- = \begin{cases} f(x)_i - 1 & i = t \\ f(x)_i + \sqrt{N} & i = a \text{ (attempt class, if specified)} \\ f(x)_i & \text{otherwise} \end{cases} \quad (4)$$

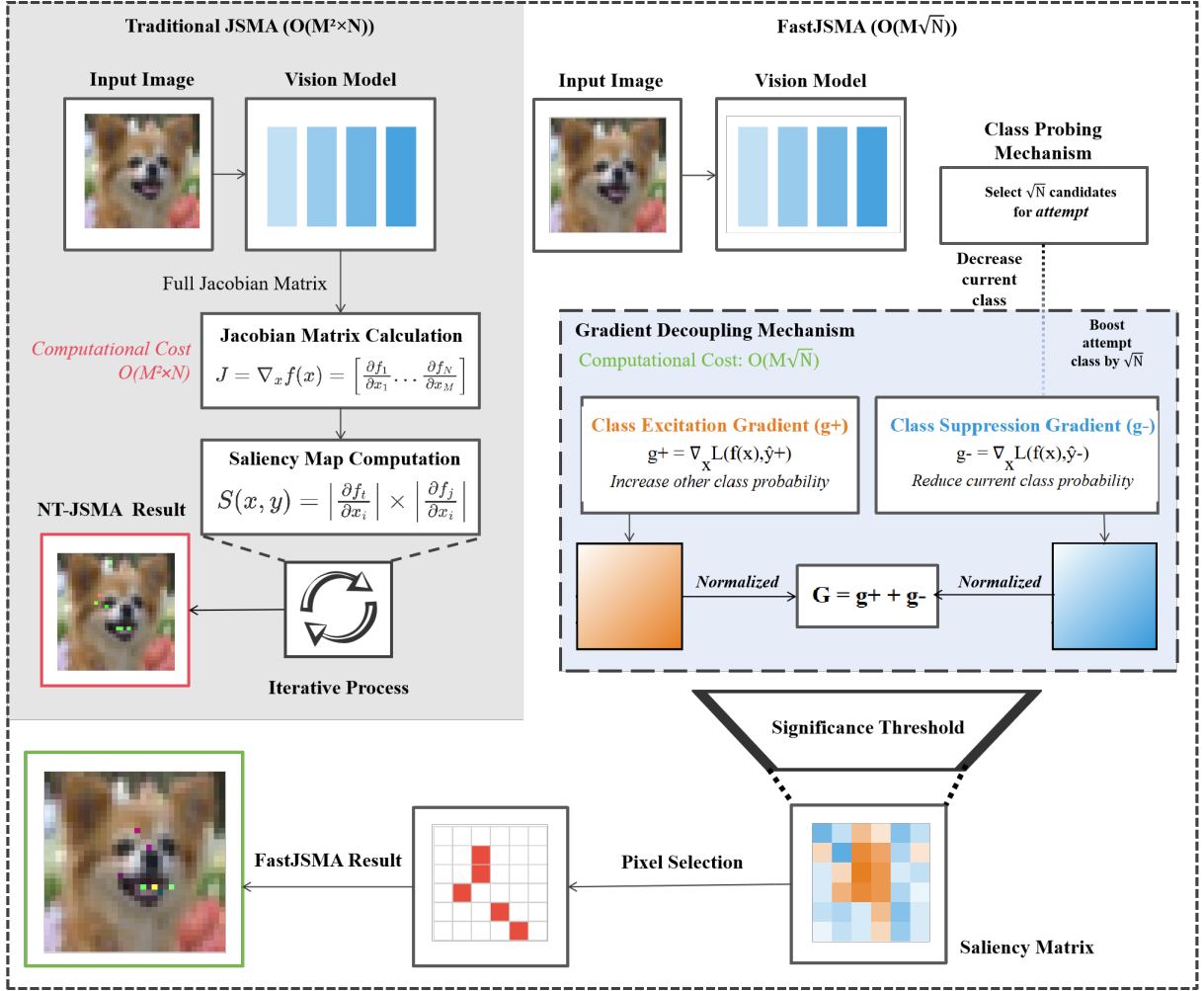


Figure 1. Architectural comparison between traditional JSMA and our proposed FastJSMA. **Left:** Traditional JSMA calculates the full Jacobian matrix with computational complexity $O(M^2 \times N)$ and relies on an iterative process to achieve misclassification, making it prohibitively expensive for large-scale datasets. **Right:** FastJSMA employs a class probing mechanism to select \sqrt{N} candidate classes and a gradient decoupling strategy that approximates the saliency computation through class suppression gradient (g^-) and class excitation gradient (g^+). The class suppression gradient reduces the current class probability while boosting an attempt class by \sqrt{N} , and the significance threshold efficiently filters the most influential pixels. This approach reduces complexity to $O(M\sqrt{N})$ while maintaining comparable effectiveness.

Calculate the gradient:

$$g^- = \nabla_x \mathcal{L}(f(x), \hat{y}^-) \quad (5)$$

This gradient reflects the direction of reducing the prediction probability of the current class and optionally enhancing a specific alternative class if an attempt parameter is provided.

- Class Excitation Gradient:** Construct an enhancement label $\hat{y}^+ \in \mathbb{R}^N$, where

$$\hat{y}_i^+ = \begin{cases} f(x)_i - 1 & i = t \\ f(x)_i + 1 & \text{otherwise} \end{cases} \quad (6)$$

Calculate the gradient:

$$g^+ = \nabla_x \mathcal{L}(f(x), \hat{y}^+) \quad (7)$$

This gradient indicates the direction of enhancing the prediction probability of other classes while suppressing the current class.

This formulation represents a significant mathematical innovation over traditional JSMA. Rather than explicitly computing $N \times M$ partial derivatives as required by the Jacobian matrix, our approach leverages automatic differentiation to compute just two gradient vectors—reducing the computational complexity by a factor proportional to the number of classes N and the number of pixels M . Impor-

tantly, this efficiency gain comes without compromising the underlying attack methodology; the dual-gradient approach preserves the core mathematical principles of JSMA while dramatically reducing its computational footprint.

The design of these two gradients cleverly avoids directly computing the Jacobian matrix while retaining the core idea of JSMA: simultaneously considering the suppression of the target class and the excitation of other classes.

3.3. Saliency Map Generation

Traditional JSMA generates saliency maps through exhaustive computation of the Jacobian matrix followed by pairwise evaluation of pixel modifications—a process that scales quadratically with the number of pixels. FastJSMA fundamentally reconceptualizes this approach through an elegant mathematical transformation that yields equivalent results with substantially reduced computation.

Based on the dual-gradient approximation, we apply gradient normalization to ensure balanced contribution while strategically emphasizing the class suppression component:

$$g_{\text{normalized}}^- = \frac{g^-}{\text{mean}(|g^-|)} \times N \quad (8)$$

$$g_{\text{normalized}}^+ = \frac{g^+}{\text{mean}(|g^+|)} \quad (9)$$

This normalization procedure includes a deliberate engineering optimization: the class suppression gradient is scaled by a factor of N (the number of classes) to amplify its influence in the overall saliency calculation. Our empirical investigation indicates that this scaling enhances attack effectiveness by emphasizing the suppression of the original class prediction.

We then integrate these normalized gradients to obtain a comprehensive gain matrix:

$$G = g_{\text{normalized}}^+ + g_{\text{normalized}}^- \quad (10)$$

This matrix captures the bidirectional influence of modifying each pixel on both decreasing the target class probability and increasing alternative class probabilities—precisely the information encapsulated in traditional JSMA’s saliency computation, but obtained through a mathematically equivalent yet computationally superior formulation.

After obtaining the gain matrix, we apply a significance threshold to filter out pixels with minimal impact. This threshold operation is visually represented as a funnel-like process in Figure 1, focusing computational resources only on the most influential pixels. By implementing this significance threshold, we effectively reduce the search space while ensuring that the selected pixels have meaningful contributions to the adversarial perturbation.

This selective modification approach, governed by both the saliency ranking and the significance threshold, ensures efficient resource allocation by concentrating perturbations on the most influential and impactful pixels.

3.4. Attack Success Rate Optimization via Class Probing

A significant enhancement in our implementation is the introduction of a class-specific probing mechanism, visually illustrated in Figure 1 as “Class Probing Mechanism”. This component substantially improves attack success rates while maintaining the untargeted nature of FastJSMA by intelligently selecting candidate classes for the attempt parameter. The optimization addresses a fundamental challenge in untargeted adversarial attacks: determining which gradient directions most efficiently induce misclassification.

Our implementation employs an intelligent candidate class exploration strategy governed by a square-root sampling heuristic:

$$\text{num_candidates} = \sqrt{N} \quad (11)$$

where N represents the number of classes in the classification problem. This square-root relationship establishes a principled balance between computational efficiency and exploration thoroughness—scaling sublinearly with the class space dimensionality to maintain tractability even for large-scale classification problems.

For each candidate class c among the top num_candidates predicted classes (excluding the original class), the algorithm:

1. Applies FastJSMA with class c as an “attempt” parameter, which specifically modifies the class suppression gradient (g^-) in two ways: - Decreases the current class probability (by setting $\hat{y}_i^- = f(x)_i - 1$ for $i = t$) - Boosts the attempt class c by a factor of \sqrt{N} (by setting $\hat{y}_i^- = f(x)_i + \sqrt{N}$ for $i = c$)
2. Evaluates whether this modified gradient direction successfully induces misclassification
3. Returns the first successful perturbation, terminating further exploration

This dual action of decreasing the current class while simultaneously boosting a specific attempt class efficiently guides the gradient toward vulnerable regions of the decision boundary, substantially improving attack success rates without requiring exhaustive exploration of all possible class transitions.

It is crucial to emphasize that this approach does not constitute a targeted attack in the conventional sense. Rather, the attempt parameter merely directs the gradient toward regions where transitions away from the original class are most probable, while maintaining an untargeted classification objective. This gradient steering mechanism represents

a methodological innovation particularly effective in high-dimensional classification spaces where traditional methods might struggle to efficiently locate vulnerable decision boundaries.

Only if all candidate-based attempts fail does the algorithm revert to the standard untargeted FastJSMA approach without class-specific gradient modification. This hierarchical strategy enhances attack success rates while maintaining the untargeted objective—an engineering optimization that improves practical performance without altering the fundamental attack paradigm.

3.5. Complexity Analysis

To theoretically analyze algorithm efficiency, we compare the computational complexity of FastJSMA with traditional JSMA. Assuming a dataset with N classes, where each image has M pixels. The computational complexity of traditional JSMA is $O(M^2 \times N)$. This is because, when constructing the saliency map, it needs to calculate the joint saliency of each pixel pair, considering M^2 order of magnitude pixel pairs; the saliency calculation for each pixel pair involves N classes. Moreover, as illustrated in Figure 1, traditional JSMA typically requires multiple iterations of this expensive computation process to achieve successful misclassification, further compounding the computational burden.

In contrast, our FastJSMA has a computational complexity of $O(M\sqrt{N})$. This is because we only perform 2 gradient calculations. The combination of gradient decoupling, class probing mechanism, and significance threshold filtering together dramatically reduces the computational requirements while maintaining attack effectiveness.

This significant reduction in complexity enables FastJSMA to efficiently process large-scale datasets. Due to the dual-gradient approximation strategy retaining JSMA’s core ideas, the attack effectiveness is preserved despite substantial complexity reduction, as verified in our experiments.

Our additional optimizations—including gradient normalization with class-specific scaling, dynamic significance thresholds, and the square-root probing strategy—further enhance efficiency without compromising mathematical foundations. These refinements demonstrate the importance of implementation details in translating theoretical innovations into practical adversarial methodologies that scale to contemporary deep learning applications.

4. Experiments

We present a comprehensive evaluation of FastJSMA, examining its effectiveness, efficiency, and transferability across multiple datasets and model architectures. Our experiments are designed to answer three key questions: (1) How does FastJSMA compare to traditional JSMA and its

variants in terms of attack success and computational efficiency? (2) How do different components of FastJSMA contribute to its overall performance? (3) How sensitive is FastJSMA to key parameter settings?

4.1. Experimental Setup

4.1.1. Implementation Environment

All experiments were conducted on a single NVIDIA RTX 3090 GPU with 24GB memory without virtual memory allocation. We implemented all methods using PyTorch framework and will release our code upon publication to ensure reproducibility.

4.1.2. Datasets and Models

To evaluate our method across varying scales of complexity, we selected three representative datasets: CIFAR-10/100 [28] (natural color images sized $32 \times 32 \times 3$, containing 10 and 100 classes respectively, each with 10,000 test samples) and ImageNet [7] (large-scale classification dataset with 1,000 classes, sized $224 \times 224 \times 3$, with 50,000 validation samples).

For model architectures, we employed widely-used ResNet variants (primarily ResNet18) [29], using pre-trained checkpoints from HuggingFace¹. These models achieve competitive performance on their respective datasets, providing a reliable foundation for our adversarial attack evaluation.

4.1.3. Attack Configuration and Baselines

We compare FastJSMA with several baseline methods:

- **JSMANT**: The traditional Jacobian-based Saliency Map Attack [5] in its non-targeted form, which computes the full Jacobian matrix.
- **WJSMA**: The Weighted JSMA variant proposed by Combey et al. [6], which introduces probabilistic weights to enhance attack effectiveness.
- **Random**: A control baseline that randomly selects pixels for modification.

To ensure fair comparison, we maintain consistent attack parameters across all methods. Since different datasets have different image sizes, we use a proportional approach to pixel modification: 8 pixels for 32×32 images (CIFAR-10/100) and 196 pixels for 224×224 images (ImageNet), representing approximately 0.78% of the total pixels in each case.

All experiments for CIFAR-10/100 were conducted on their complete test sets (10,000 samples each). For ImageNet, we used the full ImageNet1K validation set (50,000 samples across 1,000 classes), where "1K" refers to the number of classes rather than sample count. Attack Success Rate (ASR) was calculated as the percentage of sam-

¹<https://huggingface.co/edada/tocg>

Table 1. Performance comparison between FastJSMA and JSMA variants with ResNet18

Model/Dataset	Method	ASR(%)	Time(μ s)	L_2 Norm
CIFAR-10	Random	7.55	3	1.42
	FastJSMA	43.98	339	1.53
	JSMANT	43.79	11,769	1.57
	WJSMA	51.56	8,641	1.29
CIFAR-100	Random	15.31	3	1.44
	FastJSMA	45.30	751	1.75
	JSMANT	43.87	61,166	1.55
	WJSMA	49.78	51,065	1.24
ImageNet	Random	2.97	8	7.78
	FastJSMA	54.89	7,908	9.19

Note: JSMANT and WJSMA could not be tested on ImageNet due to memory constraints

ples that were successfully misclassified after perturbation out of all tested samples.

Notably, since JSMA-based methods operate iteratively (modifying two pixels per iteration), JSMANT and WJSMA use half the number of iterations compared to the total pixel modification count. FastJSMA modifies one pixel per iteration for fair comparison. The attempt parameter controls the number of candidate classes explored during class probing: we use attempt=10 for CIFAR-10, attempt= \sqrt{N} for larger datasets (e.g., attempt=10 for CIFAR-100’s 100 classes, attempt=32 for ImageNet’s 1000 classes).

4.2. Performance Analysis

4.2.1. Comparison with JSMA Variants

Table 1 presents our main experimental results, comparing FastJSMA against JSMANT, WJSMA, and the random baseline across different datasets and model architectures. We report three key metrics: Attack Success Rate (ASR), average generation time per sample, and average perturbation magnitude (L_2 norm).

The results empirically validate our theoretical complexity reduction from $O(M^2 \times N)$ to $O(M\sqrt{N})$ proposed in Section 3.4. FastJSMA achieves attack success rates comparable to or exceeding traditional JSMANT across all datasets while demonstrating remarkable computational efficiency—requiring just 2.9% of JSMANT’s processing time on CIFAR-10 (339 μ s vs. 11,769 μ s) and merely 1.2% on CIFAR-100 (751 μ s vs. 61,166 μ s). These speed-ups of 34.7 \times and 81.4 \times respectively confirm that the gradient decoupling mechanism theorized in our mathematical formulation successfully transforms the computational scaling properties of saliency-based attacks.

The efficiency advantage increases with dataset complexity, closely matching our theoretical prediction that traditional JSMA’s complexity worsens quadratically with increasing image size and class count. Most significantly, FastJSMA successfully executes on ImageNet with a rea-

Table 2. Ablation study on gradient components

Gradient Component	ASR(%)	Time(μ s)	L_2 Norm
FastJSMA (Complete)	43.98	339	1.66
G+	30.66	432	1.71
G-	35.75	443	1.66

sonable computation time of 7,908 μ s per sample, while both JSMANT and WJSMA fail due to memory constraints. This empirical finding not only validates our theoretical analysis but extends its implications—demonstrating that our mathematical reformulation enables saliency-based attacks on previously inaccessible large-scale datasets.

WJSMA achieves somewhat higher ASRs on CIFAR datasets, which aligns with our theoretical understanding of its specialized optimization mechanisms. However, its computational demands confirm our analysis that approaches relying on the full Jacobian matrix face fundamental limitations for larger-scale applications. These results demonstrate the theoretical-practical trade-off that motivated our work: FastJSMA sacrifices minimal attack effectiveness for transformative computational efficiency gains.

4.2.2. Ablation Study on Gradient Components

To investigate the contribution of different gradient components in our dual-gradient approximation strategy, we conducted an ablation study comparing three variants of FastJSMA using the CIFAR-10 dataset with ResNet18, 8 pixels modification, and attempt=10.

These results validate and refine our theoretical decomposition of the Jacobian matrix described in Section 3.2. The complete FastJSMA incorporating both gradients achieves the highest ASR (43.98%), empirically confirming that our dual-gradient formulation successfully captures the bidirectional influence encoded in traditional JSMA’s saliency computation. The class suppression gradient (g^-) alone yields a 35.75% ASR, outperforming the class excitation gradient (g^+) alone (30.66%), revealing a key theoretical insight: reducing the current class probability has a more significant impact than increasing alternative class probabilities.

This empirical finding extends our initial theoretical understanding, revealing an asymmetric contribution pattern not initially anticipated in our mathematical formulation. This discovery explains the effectiveness of our normalization procedure in Section 3.3, which amplifies the class suppression gradient by a factor of N —the ablation empirically validates this engineering optimization while simultaneously deepening our theoretical understanding. The comparable computation times across variants (339-443 μ s) further validate our complexity analysis that dual gradient computation maintains $O(M\sqrt{N})$ efficiency regardless of variant configuration.

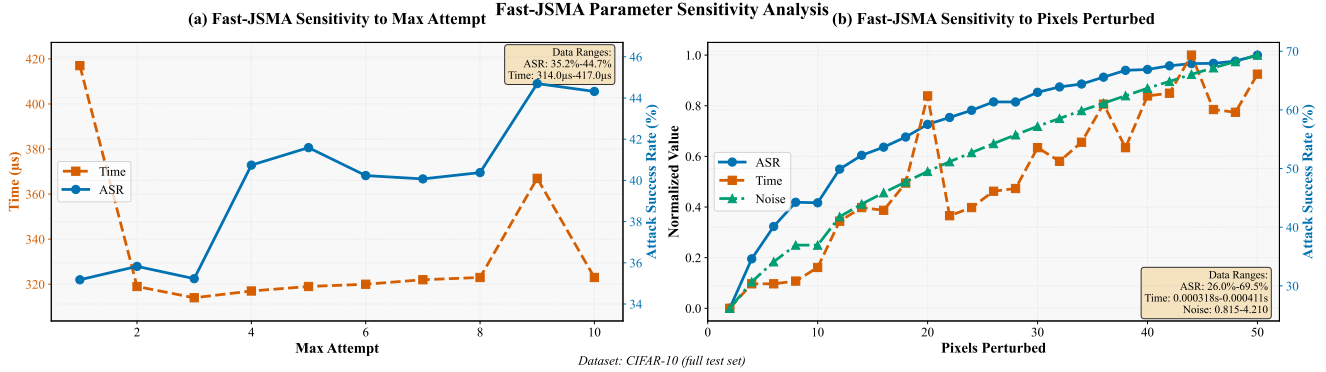


Figure 2. Parameter sensitivity analysis of FastJSMA on CIFAR-10. (a) Sensitivity to Max Attempt parameter showing ASR and Time curves. (b) Sensitivity to pixels perturbed showing ASR, Time, and Noise curves. All metrics are normalized using min-max scaling for visual comparison; actual ASR percentages shown on right y-axis. Experiments conducted on the complete CIFAR-10 test set (10,000 samples) with ResNet18.

4.2.3. Parameter Sensitivity Analysis

To evaluate FastJSMA’s robustness to parameter settings, we analyzed two key parameters: pixel modification count and the attempt parameter. Figure 2 presents these analyses conducted on the complete CIFAR-10 test set (10,000 samples).

Our Max Attempt parameter analysis (Figure 2a) shows that attack success rate follows a non-linear pattern: rapid initial growth from attempt=1 to attempt=4, followed by gradual increase thereafter. This validates our square-root sampling heuristic described in Section 3.4, confirming that smaller attempt values provide higher marginal benefits. For scalability, we empirically determined that \sqrt{N} sampling works well for larger datasets. Computational cost scales linearly with attempt value, supporting our $O(M\sqrt{N})$ complexity analysis. The data ranges from 35.2% ASR at attempt=1 to 44.7% ASR at attempt=10, with corresponding time costs increasing from $314\mu\text{s}$ to $417\mu\text{s}$.

The pixel modification analysis (Figure 2b) reveals that FastJSMA achieves substantial attack success even with minimal pixel modifications, with performance following an S-curve: initial rapid growth (2→16 pixels), moderate middle phase growth (16→36 pixels), and final diminishing returns (36→50 pixels). This validates our significance threshold mechanism in Section 3.3. Runtime increases only marginally despite significant increases in modified pixels, while L2 norm scales approximately linearly with pixel count. The analysis shows ASR ranging from 26.0% with 2 pixels to 69.5% with 50 pixels, with average generation time remaining stable around 0.0003-0.0004 seconds.

These findings both validate and extend our theoretical model. The consistent computational efficiency across different pixel modification counts demonstrates FastJSMA’s scalability, while the sharp initial gains in both analyses highlight the method’s effectiveness at identifying influen-

tial pixels with minimal computational overhead, making it particularly suitable for constrained attack scenarios.

5. Conclusion

This paper introduces FastJSMA, a computationally efficient reformulation of saliency-based adversarial attacks. By decomposing the traditional Jacobian calculation into complementary gradient components—class suppression (g^-) and class excitation (g^+)—we reduce computational complexity from $O(M^2 \times N)$ to $O(M\sqrt{N})$ while maintaining attack effectiveness. Our class probing mechanism and significance threshold further optimize performance by systematically identifying vulnerable decision boundaries and prioritizing influential pixels.

Experiments validate that FastJSMA achieves dramatic computational efficiency gains—up to 81.4× speedup on CIFAR-100 and successful execution on ImageNet where traditional JSMA fails entirely—while maintaining comparable attack success rates. The ablation study reveals the asymmetric contribution of gradient components, providing new insights that extend our theoretical understanding. These findings demonstrate that our mathematical reformulation successfully overcomes the computational barriers that have limited JSMA’s applicability in large-scale settings. Beyond immediate technical contributions, FastJSMA’s gradient decoupling principles may inform future research in computationally efficient adversarial attacks and explainable AI, where feature importance analysis at scale remains a significant challenge.

Acknowledgments

We would like to thank the HPC Platform of Huazhong University of Science and Technology for providing computational resources. We also express our gratitude to the School of Artificial Intelligence and Automation, the School of Software Engineering and the School of Electronic Information and Communications at Huazhong University of Science and Technology for their strong support throughout this research.

References

- [1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013. 1
- [2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 1, 2
- [3] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1, 2
- [4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 1
- [5] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016. 1, 2, 6
- [6] Théo Combey, António Loison, Maxime Faucher, and Hatem Hajri. Probabilistic jacobian-based saliency maps attacks. *Machine Learning and Knowledge Extraction*, 2(4): 558–578, 2020. doi: 10.3390/make2040030. 1, 2, 6
- [7] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009. 1, 6
- [8] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016. 2
- [9] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 15–26. ACM, 2017. 2
- [10] Dawei Zhou, Nannan Wang, Tongliang Liu, and Xinbo Gao. Improving adversarial training from the perspective of class-flipping distribution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025. 2
- [11] Dawei Zhou, Nannan Wang, Bo Han, Tongliang Liu, and Xinbo Gao. Defending against adversarial examples via modeling adversarial noise. *International Journal of Computer Vision*, pages 1–18, 2025.
- [12] Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. Understanding robust overfitting of adversarial training and beyond. In *International Conference on Machine Learning*, pages 25595–25610. PMLR, 2022. 2
- [13] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016. 2
- [14] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018. 2
- [15] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020. 2
- [16] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 2
- [17] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2
- [18] Chengzhi Mao, Mia Chiquier, Hao Wang, Junfeng Yang, and Carl Vondrick. Adversarial attacks are reversible with natural supervision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 661–671, October 2021. 2
- [19] Jianming Zhang and Stan Sclaroff. Saliency detection: A boolean map approach. In *2013 IEEE International Conference on Computer Vision*, pages 153–160, 2013. doi: 10.1109/ICCV.2013.26. 2
- [20] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019. 2, 3
- [21] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pages 1277–1294. IEEE, 2020. 2, 3
- [22] Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 2
- [23] Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: a few pixels make a big difference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9096, 2019. 3
- [24] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble

- adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. [3](#)
- [25] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, pages 2206–2216. PMLR, 2020. [3](#)
- [26] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pages 484–501. Springer, 2020. [3](#)
- [27] Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. *arXiv preprint arXiv:2102.12827*, 2021. [3](#)
- [28] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 4, University of Toronto, 2009. [6](#)
- [29] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. [6](#)