
MITIGATING BLACK-BOX SCORE-BASED QUERY ATTACKS VIA GRADIENT CONFUSION

Ali Javeri Javad Hezareh Radin Shahdaei AmirMahdi Sadeghzadeh*

Department of Computer Engineering, Sharif University of Technology
Tehran, Iran

ali.javeri78@sharif.edu javad.hezareh@sharif.edu

radin.shahdaei01@sharif.edu sadeghzadeh@sharif.edu

ABSTRACT

Score-based black-box adversarial attacks pose a serious threat to deployed machine learning systems by exploiting repeated model queries to craft adversarial examples, requiring neither access to model parameters nor architecture. Existing defenses predominantly rely on stochastic output perturbations, which fail to prevent determined attackers from recovering stable gradient estimates by increasing query budgets or averaging over randomness. In this work, we propose the **Gradient Confusion Defense (GCD)**, a retraining-free, deterministic test-time wrapper that actively shapes the probability vector returned to querying adversaries. By solving a constrained optimization problem grounded in the model’s Jacobian, GCD aligns the exposed logit-gradient with a target confusion direction that is maximally misaligned with the true adversarial loss gradient—increasing the query complexity of attacks without sacrificing predictive utility or top-1 accuracy. We provide rigorous theoretical analysis demonstrating that GCD provably slows gradient-estimation-based attacks and induces systematic accept/reject errors in random search-based attacks, with the confusion effect amplifying as the fraction of defended queries increases. Extensive experiments on CIFAR-10 and ImageNet across multiple architectures (ResNet-18, ResNet-50, ViT-B/16) and four state-of-the-art attacks confirm these predictions: GCD achieves state-of-the-art robustness, while incurring zero clean accuracy degradation. Our code is available at: <https://github.com/awlliom/GradientConfusion>.

Keywords Machine Learning, Adversarial Defense, Robustness, Adversarial Learning

1 Introduction

Modern machine learning systems are often deployed via remote interfaces (e.g., cloud APIs and online services), where users interact with models only through query access. In this setting, standard white-box threat assumptions (Szegedy et al., 2014; Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Carlini and Wagner, 2017; Athalye et al., 2018a; Madry et al., 2018) are frequently unrealistic, yet query-based black-box attacks remain highly effective even without access to model parameters, gradients, or training data (Ilyas et al., 2018; Al-Dujaili and O’Reilly, 2020; Andriushchenko et al., 2020; Cheng et al., 2020; Guo et al., 2019; Ilyas et al., 2019; Vo et al., 2024; Reza et al., 2025; Chen and Gu, 2020; Chen et al., 2020; Reza et al., 2023). By adaptively probing the model and observing outputs, adversaries can craft adversarial examples under practical query budgets, making black-box robustness a core requirement for secure and trustworthy deployment.

Query-based black-box attacks are commonly divided into *decision-based* attacks (Cheng et al., 2020; Chen and Gu, 2020; Chen et al., 2020; Reza et al., 2023), which rely only on predicted labels and search for the decision boundary, and *score-based* attacks (Andriushchenko et al., 2020; Al-Dujaili and O’Reilly, 2020; Guo et al., 2019; Ilyas et al., 2019; Vo et al., 2024), which use logits or class probabilities to estimate gradients via finite-difference or search-based strategies. In practice, score-based attacks are typically more query-efficient and achieve higher success rates, posing a serious risk to deployed classifiers that must return confidence scores. As a result, a large body of work (Qin et al.,

*Corresponding author

2021; Xie et al., 2018; He et al., 2019; Chen et al., 2022; Aithal and Li, 2022; Dashtbani et al., 2026; Hung-Quang et al., 2024; Wang et al., 2021) has proposed defenses, most of which introduce randomness into model responses (Qin et al., 2021; Aithal and Li, 2022; Dashtbani et al., 2026; Hung-Quang et al., 2024). While such methods can increase variance or slow optimization, they do not explicitly control the gradient signal exposed through outputs, allowing attackers to average over randomness (Athalye et al., 2018b) or increase the query budget to recover stable estimates when meaningful scores must be preserved.

In this work, we revisit black-box defense from a different perspective and ask: *What gradient signal should a model expose to an adversary?* Instead of relying on stochastic perturbations, we propose to actively manipulate the gradient information inferred from model outputs. Our goal is not merely to add noise, but to deliberately bias the gradient estimates or search directions that score-based attacks attempt to recover, steering them away from directions that reduce the classification margin.

We study the problem of defending a fixed, pretrained target classifier against query-based black-box adversaries while preserving correct predictions and usable score outputs. We introduce a deterministic test-time response wrapper that modifies the returned probability vector without altering the classifier itself or its training procedure. The core idea is *gradient confusion*: we deliberately shape the output scores so that the induced adversarial loss gradient’s direction deviates from the true adversarial loss gradient that score-based attacks seek to estimate. Using the loss gradients and Jacobian of the model’s logits, we derive an analytic target direction that misaligns the attacker’s search direction or estimated gradient, thereby increasing the number of queries required for successful adversarial example generation.

Building on a target confusion direction, we introduce the *Gradient Confusion Defense (GCD)*, which formulates a constrained optimization problem over output probabilities to explicitly control the gradient signal exposed to score-based black-box attackers. GCD enforces three principled constraints at test time: exact preservation of the top-1 prediction to maintain clean accuracy, a bounded ℓ_1 distortion on the returned probability vector to preserve the utility of confidence scores, and alignment of the exposed logit-gradient with a target confusion direction rather than the true gradient.

The main contributions of this work are threefold.

- We propose the *Gradient Confusion Defense (GCD)*, a practical, retraining-free test-time defense that explicitly controls the gradient signal exposed through model outputs in the score-based black-box threat model.
- We *theoretically* analyze how GCD forces the adversary to issue more queries to generate a successful adversarial example using gradient estimation-based and search-based attacks.
- *Extensive experiments* show that GCD substantially reduces attack success rates while preserving clean accuracy, and further validate our theoretical analysis and demonstrate the effectiveness of our defense against several state-of-the-art query-based and adaptive attacks.

2 Related Works

Score-based query attacks. Query-based black-box adversarial attacks craft adversarial examples using only repeated queries to a deployed model, without access to parameters or gradients. These attacks are typically divided into *score-based* methods (Andriushchenko et al., 2020; Al-Dujaili and O’Reilly, 2020; Guo et al., 2019; Ilyas et al., 2019; Vo et al., 2024; Reza et al., 2025; Ilyas et al., 2018; Chen et al., 2017), which leverage logits or class probabilities, and *decision-based* methods (Cheng et al., 2020; Chen and Gu, 2020; Chen et al., 2020; Reza et al., 2023), which use only predicted labels and are often less query-efficient. Early score-based attacks estimated gradients via zero-order optimization and finite-difference style approximations (Chen et al., 2017; Ilyas et al., 2018, 2019), while later work emphasized search-based strategies that can be highly query-efficient (Andriushchenko et al., 2020; Al-Dujaili and O’Reilly, 2020). Bandit-TD (Ilyas et al., 2019) frames gradient estimation as a bandit optimization problem and exploits temporal correlation of successive gradients and spatial structure in images. SquareAttack (Andriushchenko et al., 2020) performs randomized search using localized square-shaped perturbations, accepting only loss-decreasing proposals, yielding strong practical query efficiency. BruSLeAttack (Vo et al., 2024) targets sparse (ℓ_0) perturbations by searching over a reduced set of pixel locations to replace with values from a fixed synthetic image, and uses a Bayesian strategy guided by historical pixel-manipulation outcomes to improve query efficiency.

Black-box defenses. A foundational line of defense is adversarial training (AT) (Tramèr et al., 2018; Salman et al., 2019; Rice et al., 2020), which empirically improves robustness by training on adversarially perturbed samples. Extensions incorporate data augmentation (Rebuffi et al., 2021), denoising (Salman et al., 2020), and ensembling (Liu et al., 2018), but AT often remains costly and can reduce clean accuracy; moreover, many such defenses primarily target white-box threats and can be brittle under obfuscated-gradient effects (Athalye et al., 2018a). For query-based

threats, defenses commonly aim to disrupt the attacker’s ability to recover stable gradients from outputs, typically via randomization or output/logit manipulation. AAA (Chen et al., 2022) explicitly optimizes returned logits to confound the attacker by wrapping the original loss with a surrogate that preserves similar degradation behavior while inducing opposite gradient slopes, thereby steering optimization toward incorrect directions. RLS (Dashtbani et al., 2026) randomizes the effective scale of logits by multiplying them with a random coefficient, which empirically degrades gradient estimation and slows attacks. Boundary defense (Aithal and Li, 2022) injects noise into the model’s output to hinder convergence, and RND (Qin et al., 2021) injects noise at the input level, both aiming to increase the query complexity required for successful attacks.

3 Preliminaries

Let $\mathcal{X} \subseteq [0, 1]^d$ be the input space, and let $[K] = \{1, \dots, K\}$ denote the label set. A classifier $\mathcal{M} : \mathcal{X} \rightarrow \Delta^K$ outputs a probability vector $\mathcal{M}(\mathbf{x}) = \sigma(\mathbf{z}(\mathbf{x}))$, where $\mathbf{z}(\mathbf{x})$ denotes the logits and σ is the softmax function. Here, $\Delta^K = \{\mathbf{p} \in \mathbb{R}^K \mid \sum_i \mathbf{p}_i = 1\}$ denotes the probability simplex. Let $k \in [K]$ denote the true label of input $\mathbf{x}$. For simplicity, we denote $\hat{y} := \mathcal{M}(\mathbf{x})$.

Attacks and perturbations. An adversarial example $\mathbf{x}_{\text{adv}}$ for a sample $\mathbf{x}$ with corresponding label k is generated by adding a perturbation δ to $\mathbf{x}$. Typically, the norm of the perturbation δ is bounded so as not to change the ground-truth label of $\mathbf{x}$. Formally, an adversarial example is defined as $\mathbf{x}_{\text{adv}} = \mathbf{x} + \delta$, such that $\|\mathbf{x}_{\text{adv}} - \mathbf{x}\|_p \leq R$, where R is the perturbation budget, and the classifier $\mathcal{M}$ misclassifies $\mathbf{x}_{\text{adv}}$ (untargeted attack: $\arg \max_i \mathcal{M}(\mathbf{x}_{\text{adv}})_i \neq k$) or predicts a target class t (targeted attack: $\arg \max_i \mathcal{M}(\mathbf{x}_{\text{adv}})_i = t$).

There are two strategies for generating adversarial examples in the black-box setting: transfer-based (Papernot et al., 2017) and query-based (Chen et al., 2017). In the query-based setting (our focus), the adversary can adaptively query the model up to a budget Q that bounds the number of query the adversary can perform. Each query returns either (i) a score or probability vector (score-based setting) or (ii) only the predicted label (decision-based setting). We focus on *score-based* black-box attacks, as they are generally stronger than decision-based attacks and harder to defend (Ilyas et al., 2018). Query-based black-box attacks typically optimize an adversarial loss using only the model’s outputs.

Adversarial Loss. To attack the black-box target classifier $\mathcal{M}(\mathbf{x})$, the following adversarial loss function has been widely used in score-based attacks (Chen et al., 2017; Ilyas et al., 2018; Qin et al., 2021):

$$f(\mathbf{x}) := \mathcal{M}(\mathbf{x})_i - \mathcal{M}(\mathbf{x})_j, \quad (1)$$

where i and j denote the indices of the largest and second-largest components of the probability vector $\mathcal{M}(\mathbf{x})$, respectively. Decreasing $f(\mathbf{x})$ encourages the top prediction (typically the correct label) to lose confidence relative to its strongest competitor. There are two common approaches for minimizing $f(\mathbf{x})$: gradient estimation (Zero-Order Optimization) and random search.

Gradient estimation (Zero-Order Optimization) The most prominent approach for generating adversarial examples is to update the input in the opposite direction of the gradient of the adversarial loss function with respect to the input, $\nabla_{\mathbf{x}} f(\mathbf{x})$, so as to decrease the adversarial loss. Since the attacker cannot directly access the model’s parameters or the gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$, it instead estimates the gradient using only the model’s outputs. A gradient estimator that is widely used in score-based attacks (Ilyas et al., 2018; Nesterov and Spokoiny, 2017; Qin et al., 2021) is given by

$$\hat{\nabla}_{\mathbf{x}} f_{\mu}(\mathbf{x}) := \frac{1}{m} \sum_{i=1}^m \frac{f(\mathbf{x} + \mu \mathbf{u}_i) - f(\mathbf{x})}{\mu} \mathbf{u}_i, \quad (2)$$

where $\mathbf{u}_i \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ and $\mu > 0$ is a small smoothing parameter. In this setup, the adversary queries the model at points in a neighborhood around the input $\mathbf{x}$. As a result, they effectively observe a Gaussian-smoothed version of the loss function, denoted by $f_{\mu}(\mathbf{x})$. The attacker then performs Projected Gradient Descent (PGD) steps based on $\hat{\nabla}_{\mathbf{x}} f_{\mu}(\mathbf{x})$ while respecting the input perturbation constraint:

$$\mathbf{x}_{t+1} = \Pi_R \left(\mathbf{x}_t - \eta_t \hat{\nabla}_{\mathbf{x}} f_{\mu}(\mathbf{x}_t) \right), \quad (3)$$

where η_t is step size at iteration t , and Π_R denotes projection onto the ℓ_p -ball of radius R , i.e., $\|\mathbf{x}' - \mathbf{x}\|_p \leq R$. Each step of PGD requires one query to the target model. Increasing the number of steps required for PGD to converge increases the query complexity of the attack.

Random-search. Random search is an iterative method for minimizing the adversarial loss function. Let $\mu > 0$ denote the step size (i.e., the perturbation magnitude used by the search rule). At iteration t , given the current iterate $\mathbf{x}_t$, the attacker samples a search direction $\mathbf{u}_t \sim \mathcal{D}$ from a predefined distribution (e.g., Gaussian directions or structured perturbations), and evaluates the candidate $\mathbf{x}_t + \mu\mathbf{u}_t$. If this perturbation decreases the adversarial loss $f(\cdot)$ in (1), the attacker updates $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t + \mu\mathbf{u}_t$; otherwise, $\mathbf{x}_{t+1} \leftarrow \mathbf{x}_t$. This type of random-search (hill-climbing) rule is commonly used in query-based black-box attacks (Guo et al., 2019; Andriushchenko et al., 2020; Ilyas et al., 2019). Using the adversarial loss $f(\mathbf{x})$, the one-step improvement signal is defined as

$$h_t := f(\mathbf{x}_t + \mu\mathbf{u}_t) - f(\mathbf{x}_t). \quad (4)$$

If $h_t < 0$, then moving along $\mathbf{u}_t$ decreases the loss and is therefore considered a useful attack direction. Accordingly, the accepted update is given by

$$\mathbf{x}_{t+1} = \Pi_{\xi}(\mathbf{x}_t + \mathbb{I}(h_t < 0) \cdot \mu\mathbf{u}_t), \quad (5)$$

where $\mathbb{I}(\cdot)$ denotes the indicator function.

4 Gradient Confusion Defense (GCD)

In score-based black-box attacks, the adversary minimizes the loss $f(\mathbf{x})$ in (1) using only the model output $\hat{\mathbf{y}} \in \Delta^K$. Gradient Confusion Defense (GCD) disrupts this process by returning a modified output $\tilde{\mathbf{y}} \in \Delta^K$ instead of $\hat{\mathbf{y}}$, increasing the number of queries needed to succeed (potentially exceeding the budget Q). GCD constructs $\tilde{\mathbf{y}}$ from $\mathcal{M}$ and $\hat{\mathbf{y}}$ while preserving utility by keeping $\|\hat{\mathbf{y}} - \tilde{\mathbf{y}}\|_1$ small and ensuring $\arg\max_k(\hat{\mathbf{y}}_k) = \arg\max_k(\tilde{\mathbf{y}}_k)$. For analysis, we introduce a hypothetical model $\tilde{\mathcal{M}}$ with $\tilde{\mathbf{y}} = \tilde{\mathcal{M}}(\mathbf{x}) = \sigma(\tilde{\mathbf{z}}(\mathbf{x}))$, and drop the dependence on $\mathbf{x}$ in notation.

Since the adversary only sees the output of the defended target model by the gradient confusion method, their adversarial loss function transforms to:

$$\tilde{f}(\mathbf{x}) := \tilde{\mathcal{M}}(\mathbf{x})_i - \tilde{\mathcal{M}}(\mathbf{x})_j, \quad (6)$$

where i and j denote the indices of the largest and second-largest components of the probability vector $\tilde{\mathcal{M}}(\mathbf{x})$, respectively. The gradient confusion solves the following minimization problem to compute $\tilde{\mathbf{y}}$:

$$\begin{aligned} \arg \min_{\tilde{\mathbf{y}}} \quad & \langle \nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}) \rangle, \\ \text{s.t.} \quad & \|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi \quad \text{and} \quad \tilde{\mathbf{y}} \in \Delta^K \end{aligned} \quad (7)$$

where ξ is the distortion budget. The GCD minimization problem addresses two objectives. **(i) Confusion:** Minimizing $\langle \nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}) \rangle$ misaligns the adversary’s estimated gradient $\nabla_{\mathbf{x}} \tilde{f}(\mathbf{x})$ from the true gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$. This increases the query cost of gradient-estimation attacks and can mislead random-search attacks by flipping the sign of h_t . Theorems 4.3 and 4.4 in Section 4.1 provide theoretical support, and Section 5 validates GCD empirically. **(ii) Utility:** The constraint $\|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi$ (equivalently $\|\hat{\mathbf{y}} - \tilde{\mathbf{y}}\|_1 \leq \xi$) keeps $\tilde{\mathbf{y}} = \tilde{\mathcal{M}}(\mathbf{x})$ close to $\hat{\mathbf{y}} = \mathcal{M}(\mathbf{x})$, preserving predictive performance. Section 4.2 details how we solve (7).

4.1 Theoretical Analysis

In this section, we theoretically show that solving the optimization problem in (7) increases the model’s resistance to score-based black-box attacks. To incorporate the crafted loss function $\tilde{f}$, we define a randomized defended objective:

$$f_{\text{def}}(\mathbf{x}) := \lambda f(\mathbf{x}) + (1 - \lambda) \tilde{f}(\mathbf{x}), \quad \lambda \sim \text{Bernoulli}(p), \quad (8)$$

where λ is drawn independently for each query. This theoretical model allows us to formalize the effect of $\tilde{f}$ on the optimization problem faced by the adversary.

Under mild assumptions, we show that using f_{def} increases the number of steps required by gradient-estimation-based attacks and misleads search-based attacks. Consequently, it confuses the adversary and increases the number of queries required to craft an adversarial input. Moreover, we argue that setting $p \rightarrow 0$ maximizes this confusion effect.

Assumption 4.1. The loss functions f and $\tilde{f}$ are Lipschitz continuous with constants $L_0(f)$ and $L_0(\tilde{f})$, respectively.

Assumption 4.2. The loss functions f and $\tilde{f}$ are smooth, meaning their gradients are Lipschitz continuous with constants $L_1(f)$ and $L_1(\tilde{f})$, respectively.

4.1.1 Gradient Estimation-Based Attacks

Gradient estimation-based attacks approximate gradients using function evaluations (e.g., (2)) and update the input via projected gradient descent (e.g., (3)). Thus, the adversary applies (projected) gradient descent to minimize the adversarial loss. The number of queries performed by the adversary is denoted by T , which corresponds to the number of optimization steps. In theory, the adversary is considered to have succeeded in optimizing the adversarial loss function if the norm of its gradient is sufficiently small at the final iterate. Thus, to analyze the adversary's optimization procedure, we need to bound the gradient norm. In our framework, using (8), the gradient estimator is driven by f_{def} rather than f .

Theorem 4.3. *Suppose Assumption 4.1 holds. Consider zero-order optimization using the gradient estimator driven by f_{def} with smoothing parameter $\mu \leq \mu^*$, for T iterations and constant step size $\eta_t = \eta^*$. If, for all $\mathbf{x} \in \mathbb{R}^d$, it holds that $|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| \leq \epsilon$, where $\epsilon > 0$ is a small constant, then the following bound holds:*

$$\begin{aligned} & \frac{p}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f_{\mu}(\mathbf{x}_t)\|^2] + \frac{1-p}{T} \sum_{t=0}^{T-1} \mathbb{E}[\langle \nabla f_{\mu}(\mathbf{x}_t), \nabla \tilde{f}_{\mu}(\mathbf{x}_t) \rangle] \\ & \leq \sqrt{\frac{4d^3 L_0(f)^5 R}{\epsilon' T}} \left(p + (1-p) \frac{L_0(\tilde{f})^2}{L_0(f)^2} + 2p(1-p) \frac{\epsilon^2}{\epsilon'^2} \right). \end{aligned} \quad (9)$$

Here, R denotes the perturbation budget, $\epsilon' > 0$ is a small constant ensuring that f_{μ} is close to f , and the expectation is taken over the randomness of the optimization process up to iteration t .

The detailed proof of Theorem 4.3 is provided in Appendix B. This theorem shows that incorporating the function $\tilde{f}$ into the optimization process can effectively perturb the optimization landscape and mislead the gradient estimates. In particular, by optimizing the confusion term $\langle \nabla f_{\mu}(\mathbf{x}_t), \nabla \tilde{f}_{\mu}(\mathbf{x}_t) \rangle$, we can increase the overall bound and thereby improve robustness against gradient estimation attacks.

Implications of Theorem 4.3. Let

$$C(p) = \sqrt{\frac{4d^3 L_0(f)^5 R}{\epsilon'}} \left(p + (1-p) \frac{L_0(\tilde{f})^2}{L_0(f)^2} + 2p(1-p) \frac{\epsilon^2}{\epsilon'^2} \right),$$

be a constant depending on p and $C(0) \neq 0$. Rearranging the bound in (9) as:

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\|\nabla f_{\mu}(\mathbf{x}_t)\|^2] & \leq \frac{C(p)}{p\sqrt{T}} \\ & - \frac{(1-p)}{p} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E}[\langle \nabla f_{\mu}(\mathbf{x}_t), \nabla \tilde{f}_{\mu}(\mathbf{x}_t) \rangle] \end{aligned} \quad (10)$$

For adversary to succeed in optimization, they need to minimize the right hand side of the inequality below a certain threshold by increasing T which is the number of optimization steps. Thus the defender can increase the second term in the right hand side by optimizing the confusion term $\langle \nabla f_{\mu}(\mathbf{x}_t), \nabla \tilde{f}_{\mu}(\mathbf{x}_t) \rangle$ to be negative and large in magnitude. In other words, the defender needs to make the gradients of the two functions to be misaligned. This would force the attacker to use a larger number of optimization steps (queries) T to achieve the desired threshold, thus increasing the attack cost.

Choosing $p \rightarrow 0$. We can further increase the bound in (10) by choosing p close to 0. When $p \rightarrow 0$, the factor $(1-p)/p$ amplifies the contribution of the confusion term. Therefore, if the defender enforces $\langle \nabla f_{\mu}(\mathbf{x}_t), \nabla \tilde{f}_{\mu}(\mathbf{x}_t) \rangle$ to be negative and large in magnitude, the right-hand side of (10) can become arbitrarily large, effectively preventing the attacker from driving the average gradient norm below a fixed threshold within a limited number of iterations. This motivates choosing $p = 0$, i.e., relying entirely on $\tilde{f}$ in the optimization process.

4.1.2 Search-Based Attacks

Random-search score-based attacks propose a direction $\mathbf{u}_t$ and accept or reject it based on the *sign* of the one-step improvement signal (cf. (4) and (5)). Under our switching defense, each query is answered using the randomized objective f_{def} , which independently returns either f or $\tilde{f}$; thus the two evaluations used to form the improvement from $\mathbf{x}_t$ to $\mathbf{x}_t + \mu \mathbf{u}_t$ may come from *different* objectives. Let $h_{\text{def},t}$ denote the improvement signal computed from the defender's returned values. Because $h_{\text{def},t}$ can mix f and $\tilde{f}$ across the two points, $\text{sign}(h_{\text{def},t})$ can disagree with

$\text{sign}(h_t)$, causing the attacker to accept harmful steps or reject helpful ones. The next result lower bounds the probability of such one-step sign errors via (i) a geometric term depending on the angle between $\nabla f(\mathbf{x})$ and $\nabla \tilde{f}(\mathbf{x})$, and (ii) a smoothness correction scaling with μ (vanishing as $\mu \rightarrow 0$ under Assumption 4.2).

Theorem 4.4. *Let Assumption 4.2 hold. Let $d \geq 2$ and $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Fix $\mathbf{x} \in \mathbb{R}^d$ and $\mu > 0$. Define the true and defended one-step signals*

$$h := f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x}), \quad h_{\text{def}} := f_{\text{def}}(\mathbf{x} + \mu \mathbf{u}) - f_{\text{def}}(\mathbf{x}).$$

Then

$$\begin{aligned} \mathbb{P}(\text{sign}(h) \neq \text{sign}(h_{\text{def}})) &\geq \\ (1-p)^2 &\left[\frac{1}{\pi} \arccos\left(\frac{\langle \nabla f(\mathbf{x}), \nabla \tilde{f}(\mathbf{x}) \rangle}{\|\nabla f(\mathbf{x})\| \|\nabla \tilde{f}(\mathbf{x})\|} \right) \right. \\ &\left. - \frac{\mu d}{\sqrt{2\pi}} \left(\frac{L_1(f)}{\|\nabla f(\mathbf{x})\|} + \frac{L_1(\tilde{f})}{\|\nabla \tilde{f}(\mathbf{x})\|} \right) \right] \end{aligned} \quad (11)$$

The proof of Theorem 4.4 is deferred to Appendix C. The event $\text{sign}(h) \neq \text{sign}(h_{\text{def}})$ is exactly a *one-step decision error*: relative to the true objective f , the attacker either rejects a truly improving direction or accepts a non-improving direction. Theorem 4.4 shows that this error probability is lower bounded by a misalignment term, proportional to the angle between $\nabla f(\mathbf{x})$ and $\nabla \tilde{f}(\mathbf{x})$, up to a smoothness correction that decreases with the step size μ . Consequently, designing $\tilde{f}$ so that $\nabla \tilde{f}(\mathbf{x})$ is strongly misaligned with $\nabla f(\mathbf{x})$ forces random search to waste queries, degrading progress under a fixed budget. A discussion of the Gaussian direction model (and its relationship to isotropic samplers used in practice) is provided in Appendix C.3

Conclusion. In summary, this subsection provides theoretical support that GCD increases the query cost of score-based black-box attacks. Theorem 4.3 shows that inducing a negative confusion term slows gradient-estimation-based optimization, while Theorem 4.4 shows that gradient misalignment causes accept/reject errors in search-based attacks. Together, these results justify designing $\tilde{f}$ to remain close to f in value while being misaligned in gradient.

4.2 Solving the GCD Minimization Problem

GCD is a *test-time output wrapper* defense. It leaves the underlying target classifier unchanged, but modifies the probability vector returned to users. Concretely, for an input $\mathbf{x}$, the target model produces logits $\mathbf{z}(\mathbf{x}) \in \mathbb{R}^K$ and probability vector $\hat{\mathbf{y}} = \sigma(\mathbf{z}(\mathbf{x})) \in \Delta^K$. We suppose that GCD returns a perturbed distribution $\tilde{\mathbf{y}} = \sigma(\tilde{\mathbf{z}}(\mathbf{x})) \in \Delta^K$. We first focus on minimizing $\langle \nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}) \rangle$, and then incorporate the constraints $\|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi$ and $\tilde{\mathbf{y}} \in \Delta^K$.

Based on the chain rule, we can write the inner product as follows:

$$\begin{aligned} &\arg \min_{\tilde{\mathbf{y}}} \langle \nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}) \rangle \\ &= \arg \min_{\tilde{\mathbf{y}}} \langle \nabla_{\mathbf{x}} \mathbf{z}(\mathbf{x}) \nabla_{\mathbf{z}} f(\mathbf{z}(\mathbf{x})), \nabla_{\mathbf{x}} \tilde{\mathbf{z}}(\mathbf{x}) \nabla_{\tilde{\mathbf{z}}} \tilde{f}(\tilde{\mathbf{z}}(\mathbf{x})) \rangle. \end{aligned} \quad (12)$$

Solving (12) exactly is difficult because $\tilde{\mathbf{y}} = \sigma(\tilde{\mathbf{z}}(\mathbf{x}))$ appears *inside* the objective through $\tilde{f}$ and $\nabla_{\tilde{\mathbf{z}}} \tilde{f}$, so the optimal $\tilde{\mathbf{z}}$ determines the gradient terms needed to optimize for $\tilde{\mathbf{z}}$ (a circular *chicken-and-egg* dependency). Therefore, due to the constraint $\|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi$, we use a local approximation of $\tilde{\mathbf{z}}(\mathbf{x})$ by $\mathbf{z}(\mathbf{x})$ (i.e., $\mathbf{z}(\mathbf{x}) \approx \tilde{\mathbf{z}}(\mathbf{x})$), which yields $\nabla_{\mathbf{x}} \mathbf{z}(\mathbf{x}) \approx \nabla_{\mathbf{x}} \tilde{\mathbf{z}}(\mathbf{x})$. Letting $J(\mathbf{x}) = \nabla_{\mathbf{x}} \mathbf{z}(\mathbf{x})$, our objective then approximately becomes:

$$\begin{aligned} &\arg \min_{\tilde{\mathbf{y}}} \langle \nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}) \rangle \\ &\approx \arg \min_{\tilde{\mathbf{y}}} \langle J(\mathbf{x}) \nabla_{\mathbf{z}} f(\mathbf{z}(\mathbf{x})), J(\mathbf{x}) \nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x})) \rangle. \end{aligned} \quad (13)$$

Our experiments show this surrogate delivers consistent improvements against query-based black-box attacks. The only derivative that is a function of $\tilde{\mathbf{y}}$ is $\nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x}))$. Let $\mathbf{z}_k(\mathbf{x})$ denote the k -th component of the perturbed logits vector $\mathbf{z}(\mathbf{x})$. For the k -th component of $\nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x}))$, we have

$$\frac{\partial \tilde{f}(\mathbf{z}(\mathbf{x}))}{\partial \mathbf{z}_k(\mathbf{x})} = \begin{cases} \tilde{\mathbf{y}}_i(1 - \tilde{\mathbf{y}}_i) - (-\tilde{\mathbf{y}}_j \tilde{\mathbf{y}}_i), & \text{if } k = i, \\ -\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_j - \tilde{\mathbf{y}}_j(1 - \tilde{\mathbf{y}}_j), & \text{if } k = j, \\ -\tilde{\mathbf{y}}_i \tilde{\mathbf{y}}_k - (-\tilde{\mathbf{y}}_j \tilde{\mathbf{y}}_k), & \text{if } k \neq i, k \neq j. \end{cases} \quad (14)$$

where $i = \arg \max_k \tilde{\mathbf{y}}_k$ (index of predicted class after defense), and $j = \arg \max_{k \neq i} \tilde{\mathbf{y}}_k$ (index of runner-up class after defense). Since we do not know the index of the predicted class and the runner-up class of $\tilde{\mathbf{y}}$ before minimization, we use the index of the predicted class and the runner-up class of $\hat{\mathbf{y}}$ to construct $\nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x}))$. Hence, $i = \arg \max_k \hat{\mathbf{y}}_k$ and $j = \arg \max_{k \neq i} \hat{\mathbf{y}}_k$.

The optimal solution for the minimization problem (13) with respect to $\nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x}))$ is as follows:

$$\nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x})) = -\alpha J(\mathbf{x})^\top J(\mathbf{x}) \nabla_{\mathbf{z}} f(\mathbf{z}(\mathbf{x})). \quad (15)$$

where $\alpha \rightarrow \infty$. In our experiments, we found that the norm of $\nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x}))$ is not important for the success of GCD. Thus, we choose $\alpha = 1$. Since we have $J(\mathbf{x})$ and $\nabla_{\mathbf{z}} f(\mathbf{z}(\mathbf{x}))$, we can compute $\nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x}))$. Because the equation $\nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x})) = \nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x}))$ with respect to $\tilde{\mathbf{y}}$ is a nonlinear system, we cannot solve it directly to obtain $\tilde{\mathbf{y}} = \tilde{\mathcal{M}}(\mathbf{x})$. Moreover, the constraints $\|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi$ and $\tilde{\mathbf{y}} \in \Delta^K$ must also be considered.

To find $\tilde{\mathbf{y}}$, GCD employs a solution that (i) returns a valid probability vector $\tilde{\mathbf{y}} \in \Delta^K$, (ii) keeps $\tilde{\mathbf{y}} = \tilde{\mathcal{M}}(\mathbf{x})$ close to $\hat{\mathbf{y}} = \mathcal{M}(\mathbf{x})$ to satisfy the utility constraint, and (iii) preserves the predicted label of the target model ($\arg \max_k \hat{\mathbf{y}}_k = \arg \max_k \tilde{\mathbf{y}}_k$). GCD solves the following minimization problem to obtain $\tilde{\mathbf{y}}$:

$$\begin{aligned} \arg \min_{\tilde{\mathbf{y}}} \quad & -\langle \nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x})), \nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x})) \rangle \\ \text{s.t.} \quad & \|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi. \end{aligned} \quad (16)$$

The inner product in (16) encourages $\nabla_{\mathbf{z}} \tilde{f}(\mathbf{z}(\mathbf{x}))$ to align with $\nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x}))$ as much as possible, subject to the constraint $\|\mathcal{M}(\mathbf{x}) - \tilde{\mathcal{M}}(\mathbf{x})\|_1 \leq \xi$. After solving the minimization problem in (16), the resulting $\tilde{\mathbf{y}}$ is normalized by dividing by its sum to ensure that $\tilde{\mathbf{y}} \in \Delta^K$. GCD solves the minimization problem in (16) using a trust-region interior-point method (Conn et al., 2000), as implemented in the `scipy` package (Virtanen et al., 2019). In rare cases, the index of the largest component of the resulting $\tilde{\mathbf{y}}$ differs from that of $\hat{\mathbf{y}}$, which reduces the accuracy of the target model. To address this issue, GCD manually adjusts $\tilde{\mathbf{y}}$. Let $i = \arg \max_k \hat{\mathbf{y}}_k$ and $j = \arg \max_k \tilde{\mathbf{y}}_k$. If $i \neq j$, GCD adds $\tilde{\mathbf{y}}_j - \tilde{\mathbf{y}}_i$ to $\tilde{\mathbf{y}}_i$ and subtracts it from $\tilde{\mathbf{y}}_j$. This guarantees that $\arg \max_k \hat{\mathbf{y}}_k = \arg \max_k \tilde{\mathbf{y}}_k$, thereby preserving the accuracy of the target model.

5 Experiments

5.1 Experimental Setup

Datasets and Models. We evaluate our proposed method, Gradient Confusion Defense across two standard benchmark datasets: CIFAR-10 and ImageNet. To ensure comprehensive assessment, we test on multiple model architectures. For CIFAR-10, we utilize pretrained ResNet-18 models. For ImageNet, we employ Torchvision pretrained ResNet-50 and Vision Transformer (ViT-B/16). All evaluations are performed on $N = 1000$ randomly sampled test images per dataset, ensuring statistical significance while maintaining computational efficiency.

Baseline Defenses. We compare GCD against five established defense methods implemented in our experimental framework: input Random Noise Defense (iRND) (Qin et al., 2021) applies Gaussian noise with $\sigma = 0.01$ to input images. Boundary Defense (oRND) (Aithal and Li, 2022) adds Gaussian noise with $\sigma = 1.0$ to output logits. Randomized Feature Defense (RFD) (Hung-Quang et al., 2024) injects noise into intermediate features. Random Logit Scaling (RLS) (Dashtbani et al., 2026) multiplies logits by a uniformly sampled scale in $[0.5, 10]$. For Adaptive Attack on Attackers (AAA), we implement AAA-sine (Chen et al., 2022) with temperature 1, deviation 0.5, attractor interval 4, calibration weight 5, learning rate 0.1, and 100 inner iterations. We only test AAA-sine, following Chen et al. (2022) which showed AAA-linear is vulnerable to adaptive attacks.

All hyperparameters were selected based on established literature or thorough validation to ensure a fair comparison. Our defense replaces the model output with the optimized probability vector obtained from Equation (16), using an output distortion budget of $\xi = 0.5$. Since the dimension of $\tilde{\mathbf{y}}$ is large for ImageNet and the trust-region interior-point method is a second-order optimizer, minimizing Equation (16) incurs a moderate computational cost. Therefore, for ImageNet we select a subset S of 20 indices of $\tilde{\mathbf{y}}$ and optimize Equation (16) only over these components, while fixing the remaining components of $\tilde{\mathbf{y}}$ to their corresponding values in $\hat{\mathbf{y}}$. To construct S , we use $\nabla_{\mathbf{z}}^* \tilde{f}(\mathbf{z}(\mathbf{x}))$ and select the top-10 and bottom-10 components, which we found to have the largest effect on the objective in Equation (16). As a result, for ImageNet the optimized $\tilde{\mathbf{y}}$ differs from $\hat{\mathbf{y}}$ in only 20 components (before normalization).

Attacks Configuration. We evaluate defenses against four state-of-the-art score-based black-box attacks in untar-geted settings: BanditAttack (Ilyas et al., 2019) (ℓ_2 and ℓ_∞ variants), BrusleAttack (Vo et al., 2024) (ℓ_0 variant),

Table 1: Black-box attack results on CIFAR-10. ASR and average number of queries are reported.

Model	Defense	Acc. Drop	Attacks (Success % / Avg. Queries)					
			Bandit- ℓ_2	Bandit- ℓ_∞	Square- ℓ_2	Square- ℓ_∞	Sign- ℓ_∞	BrusLe- ℓ_0
ResNet-18	None	-	82.68% / 545.68	98.03% / 186.16	87.76% / 466.29	99.38% / 112.46	98.97% / 230.05	79.78% / 234.82
	iRND	1%	11.63% / 333.97	41.47% / 186.49	24.92% / 290.21	80.52% / 178.96	66.49% / 233.48	82.9% / 220.76
	oRND	0.6%	12.53% / 354.43	47.70% / 158.43	20.98% / 188.31	64.09% / 113.92	38.83% / 170.05	68.6% / 201.10
	RLS	0%	04.15% / 345.9	25.11% / 23.96	12.76% / 268.69	47.31% / 90.88	21.06% / 111.41	51.77% / 316.57
	AAA	0.1%	03.33% / 318	38.08% / 132.26	9.77% / 174.91	43.41% / 82.04	32.09% / 177.10	66.78% / 290.17
	GCD	0%	0.94% / 131.66	28.12% / 99.37	4.15% / 20.95	30.71% / 11.10	14.53% / 72.14	38.59% / 269.82

Table 2: Black-box attack results on ImageNet. ASR and average number of queries are reported.

Model	Defense	Acc. Drop	Attacks (Success % / Avg. Queries)				
			Bandit- ℓ_2	Bandit- ℓ_∞	Square- ℓ_2	Square- ℓ_∞	Sign- ℓ_∞
ResNet-50	None	-	46.73% / 761.22	87.05% / 251.79	81.45% / 483.65	99.47% / 133.90	91.33% / 328.71
	iRND	0.2%	11.02% / 440.30	45.56% / 141.79	26.66% / 214.17	83.49% / 126.10	60.78% / 229.78
	oRND	1.7%	51.47% / 229.94	75.95% / 103.83	59.84% / 208.49	83.52% / 96.91	69.47% / 182.20
	RLS	0%	6.68% / 618.42	47.80% / 143.72	16.29% / 143.03	59.82% / 104.07	21.37% / 109.13
	AAA	0.3%	11.80% / 356.96	37.94% / 85.10	15.69% / 32.05	47.06% / 33.59	23.6% / 109.37
	GCD	0%	3.34% / 528.6	35.65% / 103.29	13.22% / 64.05	47.94% / 32.17	20.17% / 90.15
ViT-B/16	None	-	38.17% / 801.41	76.46% / 319.75	70.64% / 566.65	97.90% / 351.15	80.05% / 597.98
	iRND	0.2%	7.78% / 471.24	25.13% / 169.19	21.77% / 287.48	61.70% / 252.02	38% / 255.65
	oRND	1.8%	46.71% / 266.73	69.17% / 191.07	58.88% / 291.56	97.32% / 84.39	81.44% / 313.67
	RLS	0%	4.47% / 589.5	28.51% / 224.63	10.29% / 148.42	30.36% / 144.31	13.39% / 184.20
	AAA	0%	8.17% / 300.13	22.28% / 102.86	10.9% / 43.72	26.74% / 92.46	15.1% / 236.60
	GCD	0%	3.35% / 609	21.94% / 140.03	12.38% / 455.74	33.83% / 257.21	21.44% / 315.70

SignAttack (Al-Dujaili and O'Reilly, 2020) (ℓ_∞ variant), and SquareAttack (Andriushchenko et al., 2020) (ℓ_2 and ℓ_∞ variants). For CIFAR-10, inputs are scaled to $[0, 255]$ with ℓ_∞ budget $\epsilon = 12.75$ (0.05 normalized) and ℓ_2 budget $\epsilon = 255$ (1.0 normalized). For ImageNet, inputs are in $[0, 1]$ with ℓ_∞ budget $\epsilon = 0.05$, and ℓ_2 budgets follow the original implementations (Bandit: $\epsilon = 1275$, Square: $\epsilon = 5$). All attacks are constrained by a maximum query budget of $Q = 1000$ queries per sample.

Metrics. We report three metrics for each (dataset, model, defense, attack) configuration: *Attack Success Rate*, the fraction of samples successfully attacked within the query budget; *Average Queries*, the mean queries per successful attack; and *Clean Accuracy Drop* for each defended model. Together, they capture robustness and utility, enabling evaluation of the defense-utility trade-off.

5.2 Results and Analysis

Overall Performance. Tables 1 and 2 summarize results on CIFAR-10 (ResNet-18) and ImageNet (ResNet-50, ViT-B/16). Across settings, GCD achieves a strong robustness-utility trade-off, preserving clean accuracy while reducing attack success rates. On CIFAR-10, GCD lowers Square- ℓ_2 ASR from 87.76% (undefended) to 4.15%, and Sign- ℓ_∞ from 98.97% to 14.53%. This effectiveness transfers to ImageNet: for ResNet-50, GCD incurs no clean accuracy drop while maintaining strong robustness. Results on ViT-B/16 further confirm robustness; for example, Square- ℓ_∞ ASR decreases from 97.90% to 33.83%.

Comparison with Other Defenses. Tables 1 and 2 also show that on non-transformer models, GCD outperforms nearly all baselines. While randomized defenses (iRND, oRND, RLS) can reduce attack success, they often incur accuracy drops or weaker protection; iRND and oRND trade accuracy for limited robustness. AAA and RLS preserve accuracy and can be competitive, but are more vulnerable to adaptive attacks (Section 5.3). Table 3 reports ℓ_1 deviations over 1000 ImageNet samples: GCD exhibits a larger shift, consistent with shaping scores to misalign the induced adversarial loss gradient rather than merely injecting noise.

Table 3: ℓ_1 distance between the original output of the model $\hat{y}$ and defense output $\tilde{y}$ for 1000 ImageNet samples.

Defense	Min ℓ_1	Avg ℓ_1	Max ℓ_1
iRND	0.00	0.11	1.03
oRND	0.00	0.26	1.77
RLS	0.00	0.33	1.74
AAA	0.00	0.28	1.60
GCD	0.07	0.58	1.25

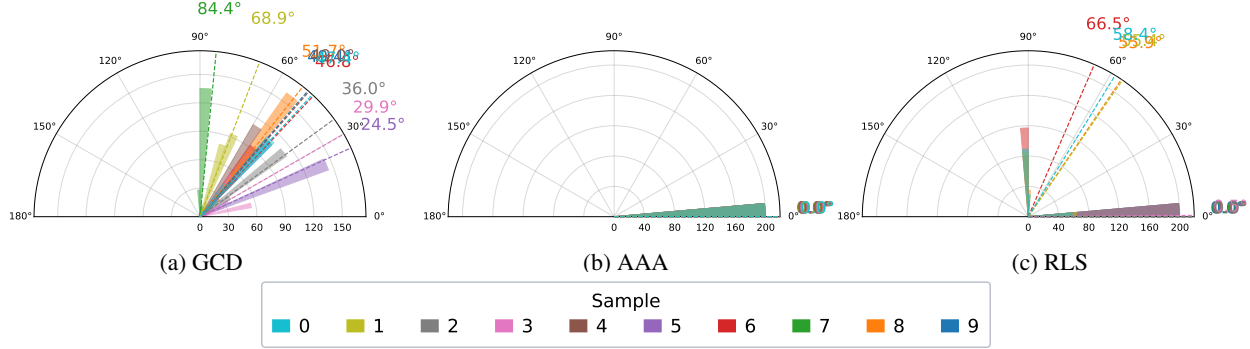


Figure 1: **Polar histograms of gradient direction misalignment.** For 10 randomly selected samples from the ImageNet test set, we plot the angular deviation $\theta = \angle(\nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}))$ between the direction $\nabla_{\mathbf{x}} \tilde{f}(\mathbf{x})$ induced by the defended score responses (i.e., the direction the attacker would follow) and the true loss gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$. Each color corresponds to 200 square- ℓ_{∞} attack iterations (200 queries used to generate adversarial example for a sample). Dashed rays and numeric labels indicate the mean deviation for each sample.

Gradient Misalignment Evidence. Figure 1 shows $\theta = \angle(\nabla_{\mathbf{x}} f(\mathbf{x}), \nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}))$, the deviation between the true loss gradient $\nabla_{\mathbf{x}} f(\mathbf{x})$ and the gradient $\nabla_{\mathbf{x}} \tilde{f}(\mathbf{x})$ implied by the defended score responses (i.e., the direction the attacker follows). GCD induces a *systematic* bias in this recovered direction, increasing the query cost of score-based attacks. In particular, GCD yields large deviations across queries (e.g., 24.5°–84.4°, many near $\sim 50^\circ$), whereas AAA concentrates at $\theta \approx 0^\circ$ (nearly revealing the true direction). RLS shows mixed behavior, including near-zero angles that still provide a reliable descent direction and angles near 90° . We hypothesize this occurs because the RLS transform maps some target-model output probability vectors to one-hot vectors, making $\nabla_{\mathbf{x}} \tilde{f}(\mathbf{x}) \approx 0$. As a result, GCD’s persistent misalignment compounds over iterations, increasing query complexity and reducing attack success, while defenses that often yield $\theta \approx 0^\circ$ preserve a low-query path. These observations support our theoretical analysis.

Figure 2 demonstrates trajectory-level analysis for 10 random CIFAR-10 samples over 100 Square Attack iterations: for each query iteration t along the attack trajectory, we measure the angle θ_t between the direction exposed under the defense and the corresponding undefended direction. This is the operational quantity that matters to an iterative score-based attacker. Based on these results, AAA does exhibit occasional local reversals, but they are not persistent: after a brief transient, its mean angle quickly settles to a small range (about 8° – 10° in our trace plot), and many sample traces remain near low angles. In contrast, GCD maintains substantially larger misalignment throughout the trajectory, with the mean staying around 25° – 28° , and many samples remaining in the 30° – 45° range over tens of queries. So the practical difference is not whether a defense can reverse a local slope at isolated points, but whether it sustains input-space misalignment along the whole optimization path. GCD does; AAA usually does not. This also explains the mechanism. AAA heuristically flips local slopes in output space, but it does not optimize the induced input-space direction after composition with the model Jacobian. GCD is Jacobian-aware: Eqs. (7), (15), and (16) explicitly optimize the exposed direction in input space. That is precisely the quantity that matters for score-based attacks. This interpretation is also consistent with our theory: Theorem 4.4 shows that larger angular misalignment increases the probability of accept/reject errors in search-based attacks, and when that larger angle is maintained across many iterations, those errors compound over the trajectory.

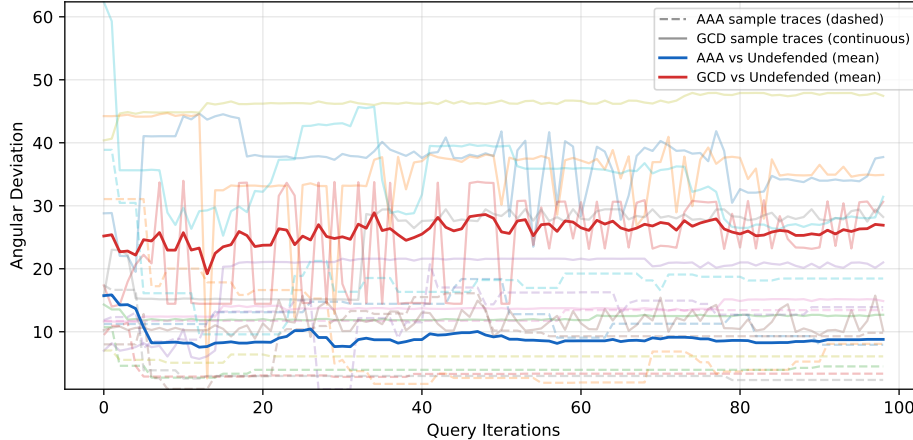


Figure 2: Angular deviation between defense-exposed and undefended gradient directions over 100 Square Attack iterations on 10 CIFAR-10 samples.

Table 4: Adaptive Attacks on GCD, RLS, and AAA. We incorporate two adaptive techniques—Expectation over Transformation (EOT) and Pendulum—into the Square Attack framework. For both adaptive variants, we set $k = 10$ and evaluate performance on 1000 ImageNet samples using ViT-B/16.

DEFENSE	EOT	PEN-SQUARE	MAX(EOT,PEN)
AAA	21.94% / 85.44	40.53% / 306.08	40.53%
RLS	36.19% / 215.83	23.55% / 85.44	36.19%
GCD	24.04% / 159.69	30.12% / 209.33	30.12%

5.3 Adaptive Attack Evaluation

To assess defense resilience against adaptive adversaries, we evaluate GCD alongside AAA and RLS against two adaptive attacks: Expectation over Transformation (EOT) (Athalye et al., 2018b) and PendulumAttack (Dashtbani et al., 2026). For sample $\mathbf{x}$, EOT counteracts random-based defenses by averaging over k independently randomized defense outputs ($\frac{1}{k} \sum_{i=0}^{k-1} \tilde{\mathcal{M}}^{(i)}$) to obtain an unbiased estimation of the $\hat{\mathbf{y}} = \mathcal{M}(\mathbf{x})$. PendulumAttack employs iterative estimation and exploitation of defense patterns.

Table 4 reveals that GCD maintains performance even under adaptive settings. The primary result is reported in the Max column, which corresponds to a worst-case adaptive threat model in which the attacker uses the more successful of the two evaluated adaptive variants. Under this setting, GCD exhibits stronger robustness than both AAA and RLS.

Figure 2 provides a mechanism-level interpretation of this result. GCD introduces a deterministic bias into the exposed objective, whereas randomization-based defenses primarily introduce variance and AAA primarily creates local slope perturbations. Adaptive averaging is designed to reduce variance; it does not remove a deterministic directional bias. For a deterministic defense, the relevant question is therefore whether the exposed direction remains persistently displaced along the attack trajectory. Figure 2 shows that GCD maintains this displacement, while AAA realigns much more quickly.

This evidence helps explain the observed adaptive comparison, although it does not by itself establish causality. Adaptive attacks may smooth local irregularities or exploit predictable patterns, but they still optimize the systematically altered objective exposed by GCD. By contrast, AAA transforms the loss into a structured oscillatory surface. Pendulum-style attacks can exploit such structure by reversing the optimization direction when progress stalls, thereby escaping local minima and maxima. The trajectory plot therefore complements Table 4, it indicates that GCD’s effect is sustained across successive queries rather than being limited to isolated local slope changes.

6 Discussion and Conclusion

Our comprehensive evaluation across two datasets, multiple model architectures, four attack methods, and two threat models demonstrates that Gradient Confusion represents an effective new paradigm for black-box defense. By

explicitly shaping the gradient information revealed to attackers, GCD Consistently lower attack success rates across all evaluated configurations. Besides, we do not degrade the clean accuracy. Also, GCD Maintains protection even against sophisticated adaptive attacks. These results establish GCD as a state-of-the-art defense against score-based black-box attacks, offering a principled alternative to randomization-based approaches. The method’s model-agnostic implementation and deterministic operation make it particularly suitable for deployment in security-critical applications where both robustness and reliable performance are paramount. However, one limitation warrants discussion. GCD requires Jacobian computations, adding computational overhead (approximately 15% inference time increase). Future work could explore alternative optimization methods and workarounds to address this obstacle.

References

- Aithal, M.B., Li, X., 2022. Boundary defense against black-box adversarial attacks, in: 26th International Conference on Pattern Recognition, ICPR 2022, Montreal, QC, Canada, August 21-25, 2022, IEEE. pp. 2349–2356.
- Al-Dujaili, A., O’Reilly, U., 2020. Sign bits are all you need for black-box attacks, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net.
- Andriushchenko, M., Croce, F., Flammarion, N., Hein, M., 2020. Square attack: A query-efficient black-box adversarial attack via random search, in: Vedaldi, A., Bischof, H., Brox, T., Frahm, J. (Eds.), Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XXIII, Springer. pp. 484–501.
- Athalye, A., Carlini, N., Wagner, D.A., 2018a. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples, in: Dy, J.G., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, PMLR. pp. 274–283.
- Athalye, A., Engstrom, L., Ilyas, A., Kwok, K., 2018b. Synthesizing robust adversarial examples, in: Dy, J.G., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, PMLR. pp. 284–293.
- Carlini, N., Wagner, D.A., 2017. Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017, IEEE Computer Society. pp. 39–57.
- Chen, J., Gu, Q., 2020. Rays: A ray searching method for hard-label adversarial attack, in: Gupta, R., Liu, Y., Tang, J., Prakash, B.A. (Eds.), KDD ’20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020, ACM. pp. 1739–1747.
- Chen, J., Jordan, M.I., Wainwright, M.J., 2020. Hopskipjumpattack: A query-efficient decision-based attack, in: 2020 IEEE Symposium on Security and Privacy, SP 2020, San Francisco, CA, USA, May 18-21, 2020, IEEE. pp. 1277–1294.
- Chen, P., Zhang, H., Sharma, Y., Yi, J., Hsieh, C., 2017. ZOO: zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in: Thuraishingham, B., Biggio, B., Freeman, D.M., Miller, B., Sinha, A. (Eds.), Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS 2017, Dallas, TX, USA, November 3, 2017, ACM. pp. 15–26.
- Chen, S., Huang, Z., Tao, Q., Wu, Y., Xie, C., Huang, X., 2022. Adversarial attack on attackers: Post-process to mitigate black-box score-based query attacks, in: Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., Oh, A. (Eds.), Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022.
- Cheng, M., Singh, S., Chen, P.H., Chen, P., Liu, S., Hsieh, C., 2020. Sign-opt: A query-efficient hard-label adversarial attack, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net.
- Conn, A.R., Gould, N.I.M., Toint, P.L., 2000. Trust Region Methods. MOS-SIAM Series on Optimization, SIAM.
- Dashtbani, H., Gandomani, M.D., Sadeghzadeh, A., 2026. Random logit scaling: Defending deep neural networks against black-box score-based adversarial example attacks. Transactions on Machine Learning Research 2026.
- Goodfellow, I.J., Shlens, J., Szegedy, C., 2015. Explaining and harnessing adversarial examples, in: Bengio, Y., LeCun, Y. (Eds.), 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.
- Guo, C., Gardner, J.R., You, Y., Wilson, A.G., Weinberger, K.Q., 2019. Simple black-box adversarial attacks, in: Chaudhuri, K., Salakhutdinov, R. (Eds.), Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, PMLR. pp. 2484–2493.

- He, Z., Rakin, A.S., Fan, D., 2019. Parametric noise injection: Trainable randomness to improve deep neural network robustness against adversarial attack, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE. pp. 588–597.
- Hung-Quang, N., Lao, Y., Pham, T., Wong, K., Doan, K.D., 2024. Understanding the robustness of randomized feature defense against query-based adversarial attacks, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net.
- Ilyas, A., Engstrom, L., Athalye, A., Lin, J., 2018. Black-box adversarial attacks with limited queries and information, in: Dy, J.G., Krause, A. (Eds.), Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, PMLR. pp. 2142–2151.
- Ilyas, A., Engstrom, L., Madry, A., 2019. Prior convictions: Black-box adversarial attacks with bandits and priors, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net.
- Liu, X., Cheng, M., Zhang, H., Hsieh, C., 2018. Towards robust neural networks via random self-ensemble, in: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (Eds.), Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII, Springer. pp. 381–397.
- Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A., 2018. Towards deep learning models resistant to adversarial attacks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings.
- Meckes, E., 2012. Approximation of projections of random vectors. Journal of Theoretical Probability 25, 333–352. doi:10.1007/s10959-010-0299-2.
- Moosavi-Dezfooli, S., Fawzi, A., Frossard, P., 2016. Deepfool: A simple and accurate method to fool deep neural networks, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society. pp. 2574–2582.
- Nesterov, Y.E., Spokoiny, V.G., 2017. Random gradient-free minimization of convex functions. Found. Comput. Math. 17, 527–566. URL: <https://doi.org/10.1007/s10208-015-9296-2>, doi:10.1007/s10208-015-9296-2.
- Papernot, N., McDaniel, P.D., Goodfellow, I.J., Jha, S., Celik, Z.B., Swami, A., 2017. Practical black-box attacks against machine learning, in: Karri, R., Sinanoglu, O., Sadeghi, A., Yi, X. (Eds.), Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017, ACM. pp. 506–519.
- Qin, Z., Fan, Y., Zha, H., Wu, B., 2021. Random noise defense against query-based black-box attacks, in: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 7650–7663.
- Rebuffi, S., Goyal, S., Calian, D.A., Stimberg, F., Wiles, O., Mann, T.A., 2021. Data augmentation can improve robustness, in: Ranzato, M., Beygelzimer, A., Dauphin, Y.N., Liang, P., Vaughan, J.W. (Eds.), Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual, pp. 29935–29948.
- Reza, M.F., Jin, R., Wu, T., Dai, H., 2025. GSBak: top-k geometric score-based black-box attack, in: The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025, OpenReview.net.
- Reza, M.F., Rahmati, A., Wu, T., Dai, H., 2023. CGBA: curvature-aware geometric black-box attack, in: IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023, IEEE. pp. 124–133.
- Rice, L., Wong, E., Kolter, J.Z., 2020. Overfitting in adversarially robust deep learning, in: Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event, PMLR. pp. 8093–8104.
- Salman, H., Li, J., Razenshteyn, I.P., Zhang, P., Zhang, H., Bubeck, S., Yang, G., 2019. Provably robust deep learning via adversarially trained smoothed classifiers, in: Wallach, H.M., Larochelle, H., Beygelzimer, A., d’Alché-Buc, F., Fox, E.B., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada, pp. 11289–11300.
- Salman, H., Sun, M., Yang, G., Kapoor, A., Kolter, J.Z., 2020. Denoised smoothing: A provable defense for pretrained classifiers, in: Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H. (Eds.), Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual.

- Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R., 2014. Intriguing properties of neural networks, in: Bengio, Y., LeCun, Y. (Eds.), 2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings.
- Tramèr, F., Kurakin, A., Papernot, N., Goodfellow, I.J., Boneh, D., McDaniel, P.D., 2018. Ensemble adversarial training: Attacks and defenses, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net.
- Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, I., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., SciPy, 2019. Scipy 1.0-fundamental algorithms for scientific computing in python. CoRR abs/1907.10121. arXiv:1907.10121.
- Vo, V.Q., Abbasnejad, E., Ranasinghe, D., 2024. Brusleattack: a query-efficient score- based black-box sparse adversarial attack, in: The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024, OpenReview.net.
- Wang, D., Ju, A., Shelhamer, E., Wagner, D.A., Darrell, T., 2021. Fighting gradients with gradients: Dynamic defenses against adversarial attacks. CoRR abs/2105.08714. arXiv:2105.08714.
- Xie, C., Wang, J., Zhang, Z., Ren, Z., Yuille, A.L., 2018. Mitigating adversarial effects through randomization, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net.

A Notations

Here one can find all notations and symbols used in the paper.

Table 5: Summary of Notations and Symbols used in the Gradient Confusion Defense (GCD) paper.

Symbol	Description
<i>General Model and Inputs</i>	
$\mathcal{X}$	Input space, $\mathcal{X} \subseteq [0, 1]^d$
d	Dimension of the input space
$[K]$	Label set $\{1, \dots, K\}$
$\mathcal{M}$	Classifier mapping $\mathcal{X} \rightarrow \Delta^K$
$\mathbf{z}(\mathbf{x})$	Logits vector output by the model for input $\mathbf{x}$
$\sigma(\cdot)$	Softmax function
Δ^K	Probability simplex $\{\mathbf{p} \in \mathbb{R}^K \mid \sum_i \mathbf{p}_i = 1\}$
$\hat{\mathbf{y}}, \mathcal{M}(\mathbf{x})$	Original probability vector output by the target model
k	True label of input $\mathbf{x}$
$J(\mathbf{x})$	Jacobian of the model's logits with respect to the input, $\nabla_{\mathbf{x}} \mathbf{z}(\mathbf{x})$
<i>Attacks and Optimization</i>	
$\mathbf{x}_{adv}$	Adversarial example ($\mathbf{x}_{adv} = \mathbf{x} + \delta$)
R	Perturbation budget (radius of the allowed perturbation ball)
Q	Maximum query budget
$f(\mathbf{x})$	Adversarial loss function based on the original model output
i, j	Indices of the largest and second-largest probability components
$\hat{\nabla}_{\mathbf{x}} f_{\mu}(\mathbf{x})$	Estimated gradient of the smoothed function
$\mathbf{u}, \mathbf{u}_t$	Random search direction vector, $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
μ	Smoothing parameter (step size for finite differences)
$f_{\mu}(\mathbf{x})$	Gaussian-smoothed version of the loss function
η_t	Step size (learning rate) at iteration t
Π_{ξ}	Projection operator onto the l_p -ball of radius ξ
h_t	One-step improvement signal in random search
<i>Gradient Confusion Defense (GCD)</i>	
$\tilde{\mathbf{y}}, \tilde{\mathcal{M}}(\mathbf{x})$	Modified/Defended probability vector returned to the user
$\tilde{\mathbf{z}}(\mathbf{x})$	Perturbed logits corresponding to $\tilde{\mathbf{y}}$
$\tilde{f}(\mathbf{x})$	Adversarial loss function based on the modified output $\tilde{\mathbf{y}}$
ξ	Distortion budget (constraint on l_1 distance between $\hat{\mathbf{y}}$ and $\tilde{\mathbf{y}}$)
$f_{\text{def}}(\mathbf{x})$	Randomized defended objective function (theoretical model)
λ	Bernoulli random variable used in the theoretical switching defense
p	Probability parameter for the Bernoulli distribution ($\lambda \sim \text{Bernoulli}(p)$)
<i>Theoretical Analysis</i>	
$L_0(f)$	Lipschitz continuity constant of function f
$L_1(f)$	Lipschitz constant of the gradient (smoothness constant) of f
T	Total number of optimization steps/queries
ϵ	Small constants bounding the difference between functions ($ f - \tilde{f} $)
ϵ'	Small constants bounding the difference between functions ($ f_{\mu} - f $)
h_{def}	Improvement signal computed from the defender's returned values

B Theoretical Analysis of Gradient Confusion for Gradient Estimation-Based Attacks

In this appendix, we provide the theoretical analysis of Theorem 4.3. We show that by incorporating a second function $\tilde{f}$, which is close to the original function f but has misaligned gradients, we can increase the convergence bound of gradient-estimation-based attacks. This analysis provides insights into how optimizing a confusion term can enhance the robustness of models against such attacks. We first state preliminaries and lemmas used in the proof, and then present the detailed proof of Theorem 4.3.

B.1 Preliminaries

Definition B.1 (μ -Gaussian Smoothing). For a function $f : \mathbb{R}^d \rightarrow \mathbb{R}$, we define its μ -Gaussian smoothed version as:

$$f_\mu(\mathbf{x}) = \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)}[f(\mathbf{x} + \mu\mathbf{u})] \quad (17)$$

Now, we define the gradient estimator as

$$g_\mu(\mathbf{x}) = \frac{f(\mathbf{x} + \mu\mathbf{u}) - f(\mathbf{x})}{\mu} \mathbf{u}, \quad (18)$$

where $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$. Using the change of variables $\mathbf{y} := \mathbf{x} + \mu\mathbf{u}$, we can show that the gradient of the smoothed function can be written as

$$\begin{aligned} \nabla f_\mu(\mathbf{x}) &= \nabla \left(\frac{1}{(2\pi)^{d/2}} \int f(\mathbf{x} + \mu\mathbf{u}) e^{-\frac{1}{2}\|\mathbf{u}\|^2} d\mathbf{u} \right) \\ &= \nabla \left(\frac{1}{\mu^{d+1}(2\pi)^{d/2}} \int f(\mathbf{y}) e^{-\frac{1}{2\mu^2}\|\mathbf{y}-\mathbf{x}\|^2} d\mathbf{y} \right) && \text{(Change of variables)} \\ &= \frac{1}{\mu^{d+1}(2\pi)^{d/2}} \int f(\mathbf{y}) \left(-\frac{\mathbf{y}-\mathbf{x}}{\mu^2} \right) e^{-\frac{1}{2\mu^2}\|\mathbf{y}-\mathbf{x}\|^2} d\mathbf{y} \\ &= \frac{1}{\mu^{d+1}(2\pi)^{d/2}} \int f(\mathbf{x} + \mu\mathbf{u}) \left(-\frac{\mu\mathbf{u}}{\mu^2} \right) e^{-\frac{1}{2}\|\mathbf{u}\|^2} \mu^d d\mathbf{u} && \text{(Change of variables)} \\ &= \frac{1}{(2\pi)^{d/2}} \int f(\mathbf{x} + \mu\mathbf{u}) \left(-\frac{\mathbf{u}}{\mu} \right) e^{-\frac{1}{2}\|\mathbf{u}\|^2} d\mathbf{u} \\ &= \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{f(\mathbf{x} + \mu\mathbf{u})}{\mu} \mathbf{u} \right] && \left(\mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{f(\mathbf{x})}{\mu} \mathbf{u} \right] = 0 \right) \\ &= \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)} \left[\frac{f(\mathbf{x} + \mu\mathbf{u}) - f(\mathbf{x})}{\mu} \mathbf{u} \right] \\ &= \mathbb{E}_{\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)}[g_\mu(\mathbf{x})] \end{aligned} \quad (19)$$

Thus, we have the following lemma:

Lemma B.2. *The gradient estimator $g_\mu(\mathbf{x})$ is an unbiased estimator of the gradient of the smoothed function $f_\mu(\mathbf{x})$, i.e., $\mathbb{E}_{\mathbf{u}}[g_\mu(\mathbf{x})] = \nabla f_\mu(\mathbf{x})$.*

We also use the following lemma on the smoothness of the Gaussian-smoothed function from Nesterov and Spokoiny (2017):

Lemma B.3. *If f is $L_0(f)$ -Lipschitz continuous, then its smoothed version f_μ is differentiable and has a Lipschitz continuous gradient with constant*

$$L_1(f_\mu) = \frac{2\sqrt{d}}{\mu} L_0(f).$$

We also use the following lemma on the moments of Gaussian random variables, which follows from standard concentration properties of Gaussian distributions:

Lemma B.4. *For $\mathbf{u} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, we have*

$$\mathbb{E}[\|\mathbf{u}\|^2] \leq d, \quad (20)$$

and

$$\mathbb{E}[\|\mathbf{u}\|^4] \leq (d+4)^2. \quad (21)$$

B.2 Bias and variance of gradient estimator

Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ and $\tilde{f} : \mathbb{R}^d \rightarrow \mathbb{R}$ be two functions such that, for all $\mathbf{x} \in \mathbb{R}^d$, $|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| \leq \epsilon$. We define a randomized objective h as

$$h(\mathbf{x}) = \lambda f(\mathbf{x}) + (1 - \lambda)\tilde{f}(\mathbf{x}), \quad (22)$$

where $\lambda \sim \text{Bernoulli}(p)$. Thus, the corresponding gradient estimator is given by

$$\begin{aligned} g_\mu(\mathbf{x}) &= \frac{(\lambda_1 f(\mathbf{x} + \mu \mathbf{u}) + (1 - \lambda_1)\tilde{f}(\mathbf{x} + \mu \mathbf{u})) - (\lambda_2 f(\mathbf{x}) + (1 - \lambda_2)\tilde{f}(\mathbf{x}))}{\mu} \mathbf{u} \\ &= \left[\lambda_1 \frac{f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x})}{\mu} + (1 - \lambda_1) \frac{\tilde{f}(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x})}{\mu} + \frac{(\lambda_1 - \lambda_2)(f(\mathbf{x}) - \tilde{f}(\mathbf{x}))}{\mu} \right] \mathbf{u} \end{aligned} \quad (23)$$

Thus, for the expectation of the gradient estimator, we have:

$$\mathbb{E}_{\lambda_1, \lambda_2} [g_\mu(\mathbf{x})] = p \left(\frac{f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x})}{\mu} \mathbf{u} \right) + (1 - p) \left(\frac{\tilde{f}(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x})}{\mu} \mathbf{u} \right) \quad (24)$$

From lemma B.2 we have:

$$\mathbb{E}_{\mathbf{u}, \lambda_1, \lambda_2} [g_\mu(\mathbf{x})] = p \nabla f_\mu(\mathbf{x}) + (1 - p) \nabla \tilde{f}_\mu(\mathbf{x}) \quad (25)$$

Now we need to compute the variance of the gradient estimator:

$$\|g_\mu(\mathbf{x})\|^2 = \underbrace{\left[\lambda_1 \frac{f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x})}{\mu} + (1 - \lambda_1) \frac{\tilde{f}(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x})}{\mu} + \frac{(\lambda_1 - \lambda_2)(f(\mathbf{x}) - \tilde{f}(\mathbf{x}))}{\mu} \right]^2}_{A} \|\mathbf{u}\|^2 \quad (26)$$

Now set $\alpha = \frac{f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x})}{\mu}$, $\tilde{\alpha} = \frac{\tilde{f}(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x})}{\mu}$, and $\delta = \frac{f(\mathbf{x}) - \tilde{f}(\mathbf{x})}{\mu}$. Thus for A we have:

$$\begin{aligned} A &= \lambda_1^2 \alpha^2 + (1 - \lambda_1)^2 \tilde{\alpha}^2 + (\lambda_1 - \lambda_2)^2 \delta^2 + 2\lambda_1(1 - \lambda_1)\alpha\tilde{\alpha} + 2\lambda_1(\lambda_1 - \lambda_2)\alpha\delta + 2(1 - \lambda_1)(\lambda_1 - \lambda_2)\tilde{\alpha}\delta \\ &\implies \mathbb{E}_{\lambda_1, \lambda_2} [A] = p\alpha^2 + (1 - p)\tilde{\alpha}^2 + 2p(1 - p)(\alpha - \tilde{\alpha})\delta + 2p(1 - p)\delta^2 \\ &= p\alpha^2 + (1 - p)\tilde{\alpha}^2 + 2p(1 - p)(\alpha - \tilde{\alpha} + \delta)\delta \\ \mathbb{E}_{\lambda_1, \lambda_2} [A] &= p \left(\frac{f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x})}{\mu} \right)^2 + (1 - p) \left(\frac{\tilde{f}(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x})}{\mu} \right)^2 \\ &\quad + \frac{2p(1 - p)}{\mu^2} (f(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x} + \mu \mathbf{u})) (f(\mathbf{x}) - \tilde{f}(\mathbf{x})) \end{aligned} \quad (27)$$

Now using the bound $|f(\mathbf{x}) - \tilde{f}(\mathbf{x})| \leq \epsilon$, and Lipschitz continuity of f and $\tilde{f}$, we have:

$$\mathbb{E}_{\lambda_1, \lambda_2} [A] \leq pL_0^2(f)\|\mathbf{u}\|^2 + (1 - p)L_0^2(\tilde{f})\|\mathbf{u}\|^2 + \frac{2p(1 - p)\epsilon^2}{\mu^2} \quad (28)$$

Putting this into equation 26, we have:

$$\mathbb{E}_{\lambda_1, \lambda_2} [\|g_\mu(\mathbf{x})\|^2] \leq pL_0^2(f)\|\mathbf{u}\|^4 + (1 - p)L_0^2(\tilde{f})\|\mathbf{u}\|^4 + \frac{2p(1 - p)\epsilon^2}{\mu^2} \|\mathbf{u}\|^2 \quad (29)$$

Now taking expectation over $\mathbf{u}$, using lemma B.4 we have:

$$\mathbb{E}_{\mathbf{u}, \lambda_1, \lambda_2} [\|g_\mu(\mathbf{x})\|^2] \leq pL_0^2(f)(d + 4)^2 + (1 - p)L_0^2(\tilde{f})(d + 4)^2 + \frac{2p(1 - p)\epsilon^2}{\mu^2} d \quad (30)$$

B.3 Descent Lemma

We now define the optimization process. Starting from $\mathbf{x}_0$, at each iteration t we sample $\lambda_1^{(t)}, \lambda_2^{(t)} \sim \text{Bernoulli}(p)$ and $\mathbf{u}^{(t)} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$, and compute the gradient estimator $g_\mu(\mathbf{x}_t)$. We then update the iterate as

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \eta_t g_\mu(\mathbf{x}_t),$$

where η_t denotes the step size at iteration t . Consequently, each iterate $\mathbf{x}_t$ is a random variable that depends on all previous samples $\{\lambda_1^{(i)}, \lambda_2^{(i)}, \mathbf{u}^{(i)}\}_{i < t}$. We define the history up to iteration t as

$$\mathcal{Z}_t = \left\{ \lambda_1^{(i)}, \lambda_2^{(i)}, \mathbf{u}^{(i)} \right\}_{i < t}.$$

Our goal is to analyze the convergence of this process in expectation over all sources of randomness. We assume the process is run for T iterations.

Having access to the variance and bias of the gradient estimator, we can now use the standard descent lemma for zero order optimization.

$$f_\mu(\mathbf{x}_{t+1}) \leq f_\mu(\mathbf{x}_t) - \eta_t \langle \nabla f_\mu(\mathbf{x}_t), g_\mu(\mathbf{x}_t) \rangle + \frac{L_1(f_\mu)}{2} \eta_t^2 \|g_\mu(\mathbf{x}_t)\|^2 \quad (31)$$

Using equations 25 and 30, and taking expectation over $\lambda_1^{(t)}, \lambda_2^{(t)}$ and $\mathbf{u}^{(t)}$ we have:

$$\begin{aligned} \mathbb{E}_{\mathbf{u}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}}[f_\mu(\mathbf{x}_{t+1})] &\leq f_\mu(\mathbf{x}_t) - \eta_t p \|\nabla f_\mu(\mathbf{x}_t)\|^2 - \eta_t (1-p) \langle \nabla f_\mu(\mathbf{x}_t), \nabla \tilde{f}_\mu(\mathbf{x}_t) \rangle \\ &\quad + \frac{L_1(f_\mu)}{2} \eta_t^2 \left(p L_0^2(f)(d+4)^2 + (1-p) L_0^2(\tilde{f})(d+4)^2 + \frac{2p(1-p)\epsilon^2}{\mu^2} d \right) \end{aligned} \quad (32)$$

Now taking expectation over the history $\mathcal{Z}_t$, we have:

$$\begin{aligned} \mathbb{E}_{\mathcal{Z}_t, \mathbf{u}^{(t)}, \lambda_1^{(t)}, \lambda_2^{(t)}}[f_\mu(\mathbf{x}_{t+1})] &= \mathbb{E}_{\mathcal{Z}_{t+1}}[f_\mu(\mathbf{x}_{t+1})] \\ &\leq \mathbb{E}_{\mathcal{Z}_t}[f_\mu(\mathbf{x}_t)] - \eta_t p \mathbb{E}_{\mathcal{Z}_t}[\|\nabla f_\mu(\mathbf{x}_t)\|^2] - \eta_t (1-p) \mathbb{E}_{\mathcal{Z}_t}[\langle \nabla f_\mu(\mathbf{x}_t), \nabla \tilde{f}_\mu(\mathbf{x}_t) \rangle] \\ &\quad + \frac{L_1(f_\mu)}{2} \eta_t^2 \left(p L_0^2(f)(d+4)^2 + (1-p) L_0^2(\tilde{f})(d+4)^2 + \frac{2p(1-p)\epsilon^2}{\mu^2} d \right) \end{aligned} \quad (33)$$

Let's define new notations for simplicity:

$$\begin{cases} A_t = \mathbb{E}_{\mathcal{Z}_t}[f_\mu(\mathbf{x}_t)] \\ B_t = \mathbb{E}_{\mathcal{Z}_t}[\|\nabla f_\mu(\mathbf{x}_t)\|^2] \\ C_t = \mathbb{E}_{\mathcal{Z}_t}[\langle \nabla f_\mu(\mathbf{x}_t), \nabla \tilde{f}_\mu(\mathbf{x}_t) \rangle] \\ D = p L_0^2(f)(d+4)^2 + (1-p) L_0^2(\tilde{f})(d+4)^2 + \frac{2p(1-p)\epsilon^2}{\mu^2} d \end{cases} \quad (34)$$

Rewrite the inequality with new notations:

$$A_{t+1} \leq A_t - \eta_t p B_t - \eta_t (1-p) C_t + \frac{L_1(f_\mu)}{2} \eta_t^2 D \quad (35)$$

Rearranging the terms, we have:

$$A_{t+1} - A_t \leq -\eta_t p B_t - \eta_t (1-p) C_t + \frac{L_1(f_\mu)}{2} \eta_t^2 D \quad (36)$$

Thus for different iterations $t = 0, \dots, T-1$, we have:

$$\begin{aligned} A_1 - A_0 &\leq -\eta_0 p B_0 - \eta_0 (1-p) C_0 + \frac{L_1(f_\mu)}{2} \eta_0^2 D \\ A_2 - A_1 &\leq -\eta_1 p B_1 - \eta_1 (1-p) C_1 + \frac{L_1(f_\mu)}{2} \eta_1^2 D \\ &\vdots \\ A_T - A_{T-1} &\leq -\eta_{T-1} p B_{T-1} - \eta_{T-1} (1-p) C_{T-1} + \frac{L_1(f_\mu)}{2} \eta_{T-1}^2 D \end{aligned} \quad (37)$$

Summing all these inequalities, we have:

$$A_T - A_0 \leq -p \sum_{t=0}^{T-1} \eta_t B_t - (1-p) \sum_{t=0}^{T-1} \eta_t C_t + \frac{L_1(f_\mu)}{2} D \sum_{t=0}^{T-1} \eta_t^2 \quad (38)$$

Here $A_0 = f_\mu(\mathbf{x}_0)$ and $A_T = \mathbb{E}_{\mathcal{Z}_T}[f_\mu(\mathbf{x}_T)]$ is the expectation of the function value at the end of the optimization process. Let's assume we use constant step size $\eta_t = \eta$ for all t . Thus we have:

$$A_T - A_0 \leq -\eta p \sum_{t=0}^{T-1} B_t - \eta(1-p) \sum_{t=0}^{T-1} C_t + \frac{L_1(f_\mu)}{2} \eta^2 DT \quad (39)$$

Taking average over T iterations, we have:

$$\frac{1}{\eta T} (A_T - A_0) \leq -p \frac{1}{T} \sum_{t=0}^{T-1} B_t - (1-p) \frac{1}{T} \sum_{t=0}^{T-1} C_t + \frac{L_1(f_\mu)}{2} \eta D \quad (40)$$

Rearranging the terms, we have:

$$p \frac{1}{T} \sum_{t=0}^{T-1} B_t + (1-p) \frac{1}{T} \sum_{t=0}^{T-1} C_t \leq \frac{A_0 - A_T}{\eta T} + \frac{L_1(f_\mu)}{2} \eta D \quad (41)$$

Now minimize the right hand side with respect to η :

$$\eta^* = \sqrt{\frac{2(A_0 - A_T)}{L_1(f_\mu)DT}} \quad (42)$$

Thus we have:

$$p \frac{1}{T} \sum_{t=0}^{T-1} B_t + (1-p) \frac{1}{T} \sum_{t=0}^{T-1} C_t \leq \sqrt{2L_1(f_\mu)D \left(\frac{A_0 - A_T}{T} \right)} \quad (43)$$

From Lipschitz continuity of f and $\tilde{f}$, we know that f_μ and $\tilde{f}_\mu$ are also Lipschitz continuous with the same constants. Thus we can bound $A_0 - A_T$ as:

$$|A_0 - A_T| \leq L_0(f) \|\mathbf{x}_0 - \mathbf{x}^*\| \leq L_0(f)R \quad (44)$$

Where R is the radius of the ball for optimization budget. Thus we have the final convergence bound as:

$$p \frac{1}{T} \sum_{t=0}^{T-1} B_t + (1-p) \frac{1}{T} \sum_{t=0}^{T-1} C_t \leq \sqrt{2L_1(f_\mu)D \left(\frac{L_0(f)R}{T} \right)} \quad (45)$$

From the definition of f_μ and lemma B.3 we have $L_1(f_\mu) = \frac{2d^{1/2}}{\mu} L_0(f)$, therefore we have:

$$p \frac{1}{T} \sum_{t=0}^{T-1} B_t + (1-p) \frac{1}{T} \sum_{t=0}^{T-1} C_t \leq \sqrt{\frac{4d^{1/2} L_0(f)^2 R}{\mu T}} D \quad (46)$$

We also need to choose μ small enough to have $|f_\mu(\mathbf{x}) - f(\mathbf{x})| \leq \epsilon'$. To have this bound, we choose $\mu \leq \frac{\epsilon'}{d^{1/2} L_0(f)}$ as in Nesterov and Spokoiny (2017). Thus we can still simplify the bound as:

$$p \frac{1}{T} \sum_{t=0}^{T-1} B_t + (1-p) \frac{1}{T} \sum_{t=0}^{T-1} C_t \leq \sqrt{\frac{4d L_0(f)^3 R}{\epsilon' T}} D \quad (47)$$

Let's also plug this value in D and rewrite that:

$$D = pL_0^2(f)(d+4)^2 + (1-p)L_0^2(\tilde{f})(d+4)^2 + \frac{2p(1-p)\epsilon^2 d^2 L_0(f)^2}{\epsilon'^2} \quad (48)$$

The input dimension d usually dominates the constant 4 in $(d+4)$, thus we can simplify the final bound as:

$$\begin{aligned} D &\approx pL_0^2(f)d^2 + (1-p)L_0^2(\tilde{f})d^2 + \frac{2p(1-p)\epsilon^2 d^2 L_0(f)^2}{\epsilon'^2} \\ &= pd^2 L_0(f)^2 \left(1 + \frac{(1-p)}{p} \frac{L_0^2(\tilde{f})}{L_0^2(f)} + 2(1-p) \frac{\epsilon^2}{\epsilon'^2} \right) \end{aligned} \quad (49)$$

Plugging this into the final bound, we have:

$$\frac{p}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\mathcal{Z}_t} [\|\nabla f_\mu(\mathbf{x}_t)\|^2] + \frac{1-p}{T} \sum_{t=0}^{T-1} \mathbb{E}_{\mathcal{Z}_t} [\langle \nabla f_\mu(\mathbf{x}_t), \nabla \tilde{f}_\mu(\mathbf{x}_t) \rangle] \leq \sqrt{\frac{4d^3 L_0(f)^5 R}{\epsilon' T} \left(p + (1-p) \frac{L_0^2(\tilde{f})}{L_0^2(f)} + 2p(1-p) \frac{\epsilon^2}{\epsilon'^2} \right)} \quad (50)$$

This bound shows that optimizing the confusion term $\langle \nabla f_\mu(\mathbf{x}), \nabla \tilde{f}_\mu(\mathbf{x}) \rangle$ increases the bound and, consequently, increases the number of steps required by gradient estimation-based attacks.

C Theoretical Analysis of Gradient Confusion for Search-Based Attacks

In this part, we provide the detailed proof of Theorem 4.4, which theoretically shows that GC can be effective against search-based attacks by misleading the acceptance/rejection signal h_t . As a result, the adversary requires more steps (and thus more queries) to craft an adversarial example. We first present auxiliary lemmas used in the proof, and then provide the detailed proof of the theorem.

C.1 Auxiliary Lemmas

Lemma C.1. Let $f : \mathbb{R}^d \rightarrow \mathbb{R}$ be L -smooth, i.e.,

$$\|\nabla f(u) - \nabla f(v)\| \leq L\|u - v\| \quad \forall u, v \in \mathbb{R}^d.$$

Then for any $x, \epsilon \in \mathbb{R}^d$,

$$f(x + \epsilon) = f(x) + \langle \nabla f(x), \epsilon \rangle + r_f,$$

where the remainder term satisfies

$$|r_f| \leq \frac{L}{2} \|\epsilon\|^2.$$

Equivalently,

$$|f(x + \epsilon) - f(x) - \langle \nabla f(x), \epsilon \rangle| \leq \frac{L}{2} \|\epsilon\|^2.$$

Proof. Define $\phi(t) = f(x + t\epsilon)$ for $t \in [0, 1]$. By the fundamental theorem of calculus,

$$f(x + \epsilon) - f(x) = \int_0^1 \langle \nabla f(x + t\epsilon), \epsilon \rangle dt.$$

Subtracting the first-order term yields

$$f(x + \epsilon) - f(x) = \langle \nabla f(x), \epsilon \rangle + \int_0^1 \langle \nabla f(x + t\epsilon) - \nabla f(x), \epsilon \rangle dt.$$

Thus the remainder term is

$$r_f = \int_0^1 \langle \nabla f(x + t\epsilon) - \nabla f(x), \epsilon \rangle dt.$$

Applying Cauchy–Schwarz and L -smoothness,

$$|r_f| \leq \int_0^1 \|\nabla f(x + t\epsilon) - \nabla f(x)\| \|\epsilon\| dt \leq \int_0^1 Lt \|\epsilon\|^2 dt = \frac{L}{2} \|\epsilon\|^2.$$

□

Lemma C.2. Let $u \sim \mathcal{N}(0, I_d)$ in $\mathbb{R}^d$ with $d \geq 2$, and let $g \in \mathbb{R}^d$ be fixed with $g \neq 0$. Then for any $c \geq 0$,

$$\mathbb{P}(|g^\top u| < c \|u\|^2) \leq \sqrt{\frac{2}{\pi}} \frac{c d}{\|g\|}.$$

Proof. Write $u = rs$ where $r = \|u\| \geq 0$ and $s = u/\|u\| \in S^{d-1}$. By rotational invariance of the standard Gaussian, r and s are independent, s is uniform on the unit sphere, and $r^2 \sim \chi_d^2$. Let $a = \|g\| > 0$. Then

$$g^\top u = r g^\top s.$$

By rotation symmetry, $g^\top s \stackrel{d}{=} aT$, where T is the first coordinate of a uniform random point on S^{d-1} . Hence

$$\mathbb{P}(|g^\top u| < c \|u\|^2 \mid r) = \mathbb{P}(|arT| < cr^2 \mid r) = \mathbb{P}\left(|T| < \frac{c}{a} r\right).$$

For $d \geq 2$, T has density on $[-1, 1]$

$$f_T(t) = C_d (1 - t^2)^{\frac{d-3}{2}}, \quad C_d = \frac{\Gamma(d/2)}{\sqrt{\pi} \Gamma((d-1)/2)}.$$

Since $\sup_{t \in [-1, 1]} f_T(t) = f_T(0) = C_d$, for any $t \geq 0$ we have

$$\mathbb{P}(|T| < t) = \int_{-t}^t f_T(x) dx \leq 2t C_d.$$

Therefore,

$$\mathbb{P}\left(|T| < \frac{c}{a} r\right) \leq 2C_d \frac{c}{a} r.$$

Taking expectation over r gives

$$\mathbb{P}(|g^\top u| < c \|u\|^2) \leq 2C_d \frac{c}{a} \mathbb{E}[r].$$

By Cauchy–Schwarz, $\mathbb{E}[r] \leq \sqrt{\mathbb{E}\|u\|^2} = \sqrt{d}$. Finally, using $\Gamma(x + \frac{1}{2})/\Gamma(x) \leq \sqrt{x}$ for all $x \geq \frac{1}{2}$ with $x = \frac{d-1}{2}$,

$$C_d = \frac{\Gamma(d/2)}{\sqrt{\pi} \Gamma((d-1)/2)} = \frac{1}{\sqrt{\pi}} \frac{\Gamma(x + \frac{1}{2})}{\Gamma(x)} \leq \sqrt{\frac{d}{2\pi}}.$$

Combining gives

$$\mathbb{P}(|g^\top u| < c \|u\|^2) \leq 2 \left(\sqrt{\frac{d}{2\pi}} \right) \frac{c}{\|g\|} (\sqrt{d}) = \sqrt{\frac{2}{\pi}} \frac{c d}{\|g\|}.$$

□

Lemma C.3. Let $u \sim \mathcal{N}(0, I_d)$ and let $g, \tilde{g} \in \mathbb{R}^d$ be nonzero vectors. Define $\theta \in [0, \pi]$ to be the angle between g and $\tilde{g}$, i.e.

$$\cos \theta = \frac{\langle g, \tilde{g} \rangle}{\|g\| \|\tilde{g}\|}.$$

Then

$$\mathbb{P}\left(\text{sign}(\langle g, u \rangle) \neq \text{sign}(\langle \tilde{g}, u \rangle)\right) = \frac{\theta}{\pi} = \frac{1}{\pi} \arccos\left(\frac{\langle g, \tilde{g} \rangle}{\|g\| \|\tilde{g}\|}\right).$$

Proof. Set $X = \langle g, u \rangle$ and $Y = \langle \tilde{g}, u \rangle$. Then (X, Y) is a centered bivariate Gaussian with

$$\text{Var}(X) = \|g\|^2, \quad \text{Var}(Y) = \|\tilde{g}\|^2, \quad \text{Cov}(X, Y) = \langle g, \tilde{g} \rangle,$$

so the correlation is $\rho = \langle g, \tilde{g} \rangle / (\|g\| \|\tilde{g}\|) = \cos \theta$. A standard orthant probability identity gives

$$\mathbb{P}(X > 0, Y > 0) = \frac{1}{4} + \frac{1}{2\pi} \arcsin(\rho).$$

By symmetry,

$$\mathbb{P}(XY < 0) = 1 - 2\mathbb{P}(X > 0, Y > 0) = \frac{1}{2} - \frac{1}{\pi} \arcsin(\rho) = \frac{1}{\pi} \arccos(\rho) = \frac{\theta}{\pi}.$$

□

C.2 Proof of Theorem 4.4

With the above lemmas in place, we now state the theorem and present a detailed proof.

Theorem C.4 (Theorem 4.4 (restated)). *Under Assumption 4.2, for $d \geq 2$, $\mathbf{u} \sim \mathcal{N}(0, I_d)$, fixed $\mathbf{x} \in \mathbb{R}^d$, $\mu > 0$, and with per-query switching defense f_{def} , the signals*

$$h := f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x}), \quad h_{\text{def}} := f_{\text{def}}(\mathbf{x} + \mu \mathbf{u}) - f_{\text{def}}(\mathbf{x})$$

satisfy the bound in (11).

Proof. Let

$$\Delta_f := f(\mathbf{x} + \mu \mathbf{u}) - f(\mathbf{x}), \quad \Delta_{\tilde{f}} := \tilde{f}(\mathbf{x} + \mu \mathbf{u}) - \tilde{f}(\mathbf{x}).$$

By definition, $h = \Delta_f$. Let $\lambda_1, \lambda_2 \stackrel{\text{i.i.d.}}{\sim} \text{Bernoulli}(p)$ be independent of u , and write

$$f_{\text{def}}(\mathbf{x} + \mu \mathbf{u}) = \lambda_1 f(\mathbf{x} + \mu \mathbf{u}) + (1 - \lambda_1) \tilde{f}(\mathbf{x} + \mu \mathbf{u}), \quad f_{\text{def}}(\mathbf{x}) = \lambda_2 f(\mathbf{x}) + (1 - \lambda_2) \tilde{f}(\mathbf{x}).$$

Define $E := \{\text{sign}(h) \neq \text{sign}(h_{\text{def}})\}$. Conditioning on (λ_1, λ_2) and keeping only the case $(0, 0)$ gives

$$\begin{aligned}\mathbb{P}(E) &\geq \mathbb{P}(E \cap \{\lambda_1 = 0, \lambda_2 = 0\}) \\ &= \mathbb{P}(\lambda_1 = 0, \lambda_2 = 0) \mathbb{P}(\text{sign}(\Delta_f) \neq \text{sign}(\Delta_{\tilde{f}})) \\ &= (1-p)^2 \mathbb{P}(\text{sign}(\Delta_f) \neq \text{sign}(\Delta_{\tilde{f}})).\end{aligned}$$

It remains to lower bound $\mathbb{P}(\text{sign}(\Delta_f) \neq \text{sign}(\Delta_{\tilde{f}}))$.

Define the linear parts

$$A := \langle \nabla f(x), u \rangle, \quad \tilde{A} := \langle \nabla \tilde{f}(x), u \rangle.$$

By Lemma C.1 applied to f with $L = L_1(f)$ and $\epsilon = \mu u$,

$$\Delta_f = \mu A + r_f, \quad |r_f| \leq \frac{L_1(f)}{2} \mu^2 \|u\|^2.$$

Similarly, applying Lemma C.1 to $\tilde{f}$ with $L = L_1(\tilde{f})$,

$$\Delta_{\tilde{f}} = \mu \tilde{A} + r_{\tilde{f}}, \quad |r_{\tilde{f}}| \leq \frac{L_1(\tilde{f})}{2} \mu^2 \|u\|^2.$$

Step 1: sign-stability events. If $|\delta| < |a|$, then $\text{sign}(a + \delta) = \text{sign}(a)$. Define

$$G_f := \left\{ \frac{L_1(f)}{2} \mu \|u\|^2 < |A| \right\}, \quad G_{\tilde{f}} := \left\{ \frac{L_1(\tilde{f})}{2} \mu \|u\|^2 < |\tilde{A}| \right\}.$$

On G_f , we have $|r_f| < \mu |A|$, hence $\text{sign}(\Delta_f) = \text{sign}(A)$; similarly on $G_{\tilde{f}}$, $\text{sign}(\Delta_{\tilde{f}}) = \text{sign}(\tilde{A})$. Therefore,

$$\mathbb{P}(\text{sign}(\Delta_f) \neq \text{sign}(\Delta_{\tilde{f}})) \geq \mathbb{P}(\text{sign}(A) \neq \text{sign}(\tilde{A})) - \mathbb{P}(G_f^c) - \mathbb{P}(G_{\tilde{f}}^c).$$

Step 2: linear sign disagreement. By Lemma C.3 with $g = \nabla f(x)$ and $\tilde{g} = \nabla \tilde{f}(x)$,

$$\mathbb{P}(\text{sign}(A) \neq \text{sign}(\tilde{A})) = \frac{1}{\pi} \arccos \left(\frac{\langle \nabla f(x), \nabla \tilde{f}(x) \rangle}{\|\nabla f(x)\| \|\nabla \tilde{f}(x)\|} \right).$$

Step 3: bound Taylor-failure probabilities. Using Lemma C.2 with $g = \nabla f(x)$ and $c = \frac{L_1(f)}{2} \mu$,

$$\mathbb{P}(G_f^c) = \mathbb{P}\left(|A| < \frac{L_1(f)}{2} \mu \|u\|^2\right) \leq \sqrt{\frac{2}{\pi}} \frac{\left(\frac{L_1(f)}{2} \mu\right) d}{\|\nabla f(x)\|} = \frac{\mu d}{\sqrt{2\pi}} \frac{L_1(f)}{\|\nabla f(x)\|}.$$

Similarly,

$$\mathbb{P}(G_{\tilde{f}}^c) \leq \frac{\mu d}{\sqrt{2\pi}} \frac{L_1(\tilde{f})}{\|\nabla \tilde{f}(x)\|}.$$

Step 4: combine and multiply by $(1-p)^2$. Combining Steps 1–3 yields

$$\mathbb{P}(\text{sign}(\Delta_f) \neq \text{sign}(\Delta_{\tilde{f}})) \geq \frac{1}{\pi} \arccos \left(\frac{\langle \nabla f(x), \nabla \tilde{f}(x) \rangle}{\|\nabla f(x)\| \|\nabla \tilde{f}(x)\|} \right) - \frac{\mu d}{\sqrt{2\pi}} \left(\frac{L_1(f)}{\|\nabla f(x)\|} + \frac{L_1(\tilde{f})}{\|\nabla \tilde{f}(x)\|} \right).$$

Multiplying by $(1-p)^2$ gives (11). $\square$

C.3 Why Gaussian Directions?

We model the search direction as Gaussian because it is rotationally invariant and makes the analysis depend only on geometry. This is also a good approximation for many isotropic samplers used in practice: in high dimensions, one-dimensional projections (i.e., inner products with a random direction) of a broad class of high-dimensional random vectors are close to Gaussian under mild conditions, so Gaussian modeling captures the sign behavior of these projections well (Meckes, 2012). In the proof, the key quantity is the probability that the two linearized one-step signals induced by $\nabla f(\mathbf{x})$ and $\nabla \tilde{f}(\mathbf{x})$ disagree in sign, and rotational invariance implies this probability depends only on their relative orientation. This is exactly where the arccosine term in (11) comes from.