Rewriting the Budget: A General Framework for Black-Box Attacks Under Cost Asymmetry

Mahdi Salmani[1] **Alireza Abdollahpoorrostam**[2] **Seyed-Mohsen Moosavi-Dezfooli**[3]
[1]University of Southern California [2]EPFL [3]Apple
salmanis@usc.edu alireza.abdollahpoorrostam@epfl.ch smoosavi@apple.com

Abstract

Traditional decision-based black-box adversarial attacks on image classifiers aim to generate adversarial examples by slightly modifying input images while keeping the number of queries low, where each query involves sending an input to the model and observing its output. Most existing methods assume that all queries have equal cost. However, in practice, queries may incur *asymmetric costs*; for example, in content moderation systems, certain output classes may trigger additional review, enforcement, or penalties, making them more costly than others. While prior work has considered such asymmetric cost settings, effective algorithms for this scenario remains underdeveloped. In this paper, we propose a general framework for decision-based attacks under asymmetric query costs, which we refer to as *asymmetric black-box attacks*. We modify two core components of existing attacks: the *search strategy* and the *gradient estimation* process. Specifically, we propose *Asymmetric Search (AS)*, a more conservative variant of binary search that reduces reliance on high-cost queries, and *Asymmetric Gradient Estimation (AGREST)*, which shifts the sampling distribution to favor low-cost queries. We design efficient algorithms that minimize total attack cost by balancing different query types, in contrast to earlier methods such as *stealthy attacks* that focus only on limiting expensive (high-cost) queries. Our method can be integrated into a range of existing black-box attacks with minimal changes. We perform both theoretical analysis and empirical evaluation on standard image classification benchmarks. Across various cost regimes, our method consistently achieves lower total query cost and smaller perturbations than existing approaches, with improvements of up to 40% in some settings. The code for *Asymmetric Attacks* is available at github.com/mahdisalmani/Asymmetric-Attacks.

1 Introduction

Decision-based adversarial attacks, first introduced by [1], generate adversarial examples in black-box settings by systematically querying a classifier and observing only its output decisions for perturbed inputs. The original Boundary Attack [1] initially required over 100,000 queries to reliably identify minimal adversarial perturbations for large-scale datasets such as ImageNet [2]. Subsequent works [3, 4, 5, 6, 7, 8] significantly enhanced the efficiency of decision-based attacks by reducing the number of queries needed, achieving improvements of one to three orders of magnitude. These advancements have led to more practical and efficient frameworks for adversarial testing in limited-query settings.

While prior work (discussed in detail in App. A) has primarily assumed that all queries have equal cost and focused on minimizing the total number of queries, in many practical scenarios, queries can incur asymmetric costs depending on their nature. For instance, Not Safe for Work (NSFW) image detection models have become increasingly important, with major platforms such as Facebook [9] and 𝕏 (formerly Twitter)[10] deploying automated mechanisms for identifying sensitive content, alongside commercial APIs developed by Google[11], Amazon [12], and Microsoft [13]. In these

Preprint. Under review.

settings, submitting explicit or borderline explicit queries could trigger more severe consequences, such as account suspension or content flagging, compared to benign queries. As a result, minimizing only the total number of queries is insufficient; effective attack strategies must account for the asymmetric costs associated with different types of queries.

Debenedetti et al. [14] introduced ***stealthy attack*** techniques to better handle asymmetric query costs. They empirically demonstrated that the standard binary search procedure for boundary point detection, mostly for projecting an adversarial point onto the decision boundary or for OPT-style gradient estimation[5], leads to a large number of high-cost queries. In particular, it can be observed from Fig.1 that at least 50% of the queries made during these attacks are high-cost. To address this, they replaced the binary search with a search strategy inspired by the classic egg dropping problem [15], which is similar to a line search algorithm. However, they did not provide a *stealthy* variant of the Monte Carlo gradient estimation used in HSJA [3], GeoDA [8], and qFool [7]. Instead, they substituted it with an OPT-style gradient estimation procedure [5].

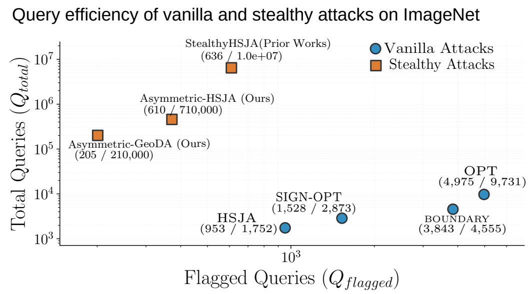

Figure 1: Each point represents the median number of queries required by an attack method to reach a median ℓ_2 norm of 10. The x-axis shows the number of flagged queries (Q_{flagged}) and the y-axis reports the total number of queries (Q_{total}). It demonstrates the superiority of our method in achieving a more favorable trade-off between flagged and total number of queries in stealthy attack settings.

Although stealthy attacks move toward addressing asymmetric query costs, they are not designed to handle arbitrary cost ratios. They implicitly assume that benign queries have zero cost, which may not reflect realistic settings where even benign queries contribute to the overall cost. In addition, since stealthy attacks could not adapt the Monte Carlo gradient approximation used in HSJA [3], they instead rely on a suboptimal and inefficient OPT-style gradient estimation [5], which is already significantly outperformed by the HSJA gradient approximation under ***symmetric*** cost settings. These limitations motivate us to answer the following question:

> **Q**: How can we develop an ***efficient*** framework to adapt attacks for any ***arbitary*** cost ratio ***without discarding*** any of their core components, including gradient estimation and binary search?

In this work, we propose a general framework for decision-based attacks that operates under arbitrary query cost asymmetries. Instead of minimizing only the number of high-cost queries, we adapt the core components of black-box attacks, namely search along adversarial paths and gradient estimation, to explicitly minimize the *total query cost*. Our framework handles any cost ratio between high-cost and low-cost queries and completely outperforms stealthy attacks by optimizing the attack structure without sacrificing efficiency. Unlike stealthy attacks [14], which modify the core mechanics of existing attacks and rely on inefficient gradient approximations, we retain the more efficient Monte Carlo gradient estimation technique used in HSJA [3], GeoDA [8], and qFool [7] with only slight adaptations to account for asymmetric costs.

We first adapt the binary search procedure to account for arbitrary query costs. Instead of splitting the search interval into two equal segments at each iteration, as in standard binary search, we take a more conservative approach by splitting the interval using the cost ratio between the high-cost and low-cost queries. This splitting strategy minimizes the expected cost, rather than simply minimizing the expected number of queries. We refer to this procedure as **A**symmetric **S**earch (AS).

Second, we adapt the gradient estimation procedure to account for arbitrary query costs. Instead of making queries around a norm ball centered at a boundary point, where approximately half of the queries are high-cost and half are low-cost as in standard HSJA, we shift the center to a point in the low-cost region and generate queries around it (Fig. 2). This adjustment intuitively reduces the frequency of high-cost queries, and the amount of shifting provides a natural way to control this frequency. To account for the shift, we weight high-cost and low-cost queries differently when estimating the gradient. We refer to this adaptation as **A**symmetric **GR**adient **EST**imation (AGREST). Our proposed framework is broadly compatible with a wide range of state-of-the-art decision-based attacks, including HSJA [3], GeoDA [8], CGBA [16]. It can be seamlessly applied to existing attack pipelines without requiring major structural changes. Through both theoretical analysis and rigorous experimental evaluation, we demonstrate that our method consistently outperforms existing attacks across arbitrary cost ratios. In particular,

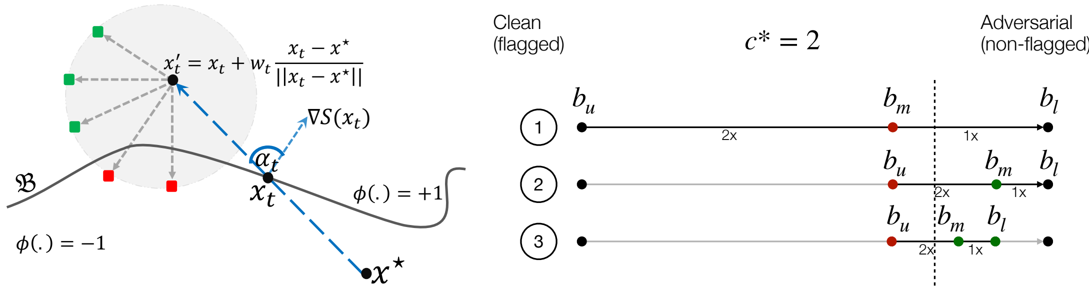

Figure 2: **Left.** Illustration of Asymmetric Gradient Estimation (AGREST), which reduces the frequency of high-cost queries by shifting the sampling region from $\mathbf{x}_t$ toward the adversarial region $\mathbf{x}'_t$ and appropriately reweighting the outcomes. **Right.** Three steps of Asymmetric Search (AS) along the path from a clean (flagged) source image to an adversarial (non-flagged) image. Flagged queries are shown in red, non-flagged queries in green, and the dashed line denotes the decision boundary.

even under extreme asymmetry conditions where the cost of high-cost queries approaches infinity, our method incurs significantly less total query cost to achieve a given adversarial perturbation size compared to prior stealthy methods [14] (see Fig. 1 (left)). This robustness highlights the effectiveness of our framework in balancing query efficiency and perturbation quality across diverse attack scenarios.

The contribution of our paper is as follows:

- To the best of our knowledge, we are the first to propose a versatile framework capable of handling arbitrary query cost ratios, providing flexibility across a wide range of adversarial attack scenarios.
- Our framework introduces **AS** and **AGREST** as two core operations, optimizing the efficiency and effectiveness of adversarial attacks.
- We provide a comprehensive theoretical analysis of the framework, establishing its foundations and demonstrating its robustness in diverse attack conditions.
- We validate the framework through extensive empirical testing on benchmark datasets and models, including ImageNet, as well as advanced models such as CLIP and Vision Transformers and ResNet. This validation highlights the framework's superior performance.

2 Problem statement

An insight into unequal queries. Consider an attacker trying to deceive an NSFW detector using decision-based methods. It may seem sufficient to choose an existing attack algorithm and add a small perturbation to an NSFW image based on that algorithm. However, this approach may encounter some practical obstacles. Most social networks enforce policies against uploading adult content, suspending users for violating these terms multiple times [10]. Using the terminology from [14], this means that the cost of queries identified by the detector as NSFW, i.e., **flagged queries**, is higher than that of other queries, i.e., **non-flagged queries**. For example, on $\mathbb{X}$, an attacker can make up to 2,400 posts per day on a single account [17]. However, after about 5 to 10 rule violations for uploading flagged posts, the attacker's account will be suspended, requiring them to create a new one. On the other hand, in existing decision-based attacks, approximately half of the made queries are flagged [14]. Therefore, if we assume the violation limit is 10, an attacker will be banned on $\mathbb{X}$ after about 20 posts. This example demonstrates the potential asymmetry in the costs of queries in a decision-based black-box setup. [14] addressed this asymmetry in costs by proposing stealthy attacks[1] designed to reduce the number of flagged queries. However, they overlooked the cost of non-flagged queries in their framework, leading to the generation of millions of non-flagged queries for every hundred flagged queries in stealthy attacks, which can also be costly. For example, in the NSFW detector scenario, assume the attacker must create a new account after reaching the daily post limit. In stealthy attacks like HSJA, the attack can generate around 10^6 non-flagged queries for every

[1]Hereafter, we refer to prior stealthy attacks simply as *stealthy attacks*, and to our approaches as *asymmetric attacks* to emphasize their cost-aware design. Though inherently stealthy due to *query cost awareness*, we adopt the term *asymmetric* attacks to distinguish our method from prior work [14].

100 flagged queries (Fig. 1). While 100 flagged queries may lead to the creation of 10 new accounts, those 10^6 non-flagged queries result in approximately 400 new accounts. This shows that non-flagged queries, despite being lower-cost, have a greater overall impact. Therefore, it is essential to develop generalized decision-based attacks that can effectively manage asymmetric query costs, making full use of low-cost queries without relying heavily on expensive ones.

General formulation. Assume that $f : \mathbb{R}^d \to \mathbb{R}^L$ is a pre-trained classifier with L classes and parameters θ. For an input image $\mathbf{x} \in [0,1]^d$, $f_k(\mathbf{x})$, the k^{th} component of $f(\mathbf{x})$, represents the predicted probability of the k^{th} class. Additionally, for each correctly classified image $\mathbf{x}$ and query image $\mathbf{x}'$, we define:

$$S_{\mathbf{x}}(\mathbf{x}') = \underset{k \neq \hat{y}(\mathbf{x})}{\arg\max} f_k(\mathbf{x}') - f_{\hat{y}(\mathbf{x})}(\mathbf{x}') \quad \text{and} \quad \phi_{\mathbf{x}}(\mathbf{x}') = \operatorname{sign}(S_{\mathbf{x}}(\mathbf{x}'))$$

Given a correctly classified source image $\mathbf{x}^\star$, the attacker's goal is to find the closest perturbed image $\mathbf{x}'$ to the source image $\mathbf{x}^\star$ such that $\phi_{\mathbf{x}^\star}(\mathbf{x}') = 1$:

$$\underset{\mathbf{x}'}{\text{minimize}} \ \|\mathbf{x}^\star - \mathbf{x}'\| \quad \text{s.t.} \quad \phi_{\mathbf{x}^\star}(\mathbf{x}') = 1 \tag{1}$$

Note that in a decision-based black-box setup, the attacker has access to $\phi_{\mathbf{x}^\star}(\mathbf{x}')$ but not $S_{\mathbf{x}^\star}(\mathbf{x}')$. Previous methods sought to solve Eq. (1) while keeping the total number of queries as low as possible. However, as discussed, asymmetric query costs can make this approach ineffective. Instead, we have to keep the total cost of queries in an asymmetric setup as low as possible:

$$\text{total cost} = N_{\text{source-class}} \cdot c^* + N_{\text{non-source-class}} \cdot 1$$

Where $N_{\text{source-class}}$ for an attack is the number of queries $\mathbf{x}'$ for which $\phi(\mathbf{x}') = -1$, $c^\star$ is the cost associated with this type of query, and $N_{\text{non-source-class}}$ is the number of queries $\mathbf{x}'$ with a cost of 1 for which $\phi(\mathbf{x}') = 1$. Existing decision-based attacks assume $c^\star = 1$, while stealthy attacks assume $c^\star = \infty$. Our goal in this paper is to propose a framework that is effective for any arbitrary value of $c^\star$, unlike vanilla and stealthy attacks. In addition, we demonstrate that our approach outperforms stealthy attacks, even when $c^\star = \infty$.

For brevity, in this paper, we omit $\mathbf{x}^\star$ when mentioning S and ϕ. Furthermore, we assume $c^\star \geq 1$, as our proposed framework is almost the same for $c^\star < 1$. This allows us to refer to queries $\mathbf{x}'$ where $\phi(\mathbf{x}') = -1$ as **high-cost queries** and others as **low-cost queries**. These new terms reflect the concept of general asymmetric costs better than the previous terms used by [14], i.e., flagged and non-flagged queries.

3 Proposed method

Decision-based black-box attacks typically involve two core operations, often applied iteratively to find small adversarial perturbations: 1. choosing a path, either straight, like GeoDA and HSJA [8, 3], or curved, like SurFree and CGBA [18, 16], and then 2. Searching along this path to find a new adversarial example, $\mathbf{x}_{t+1}$, that is closer to $\mathbf{x}^\star$ than $\mathbf{x}_t$, the adversarial example from the previous iteration. These attacks either choose a path randomly, as in Boundary Attack [1] and SurFree, or use queries to find a path that leads to a closer adversarial example than a random path, as in HSJA, GeoDA, and CGBA. To find this better-than-random path, these attacks estimate the normalized gradient direction of S at $\mathbf{x}_t$ by approximating $\nabla S(\mathbf{x}_t)$ as follows:

$$\widetilde{\nabla S}(\mathbf{x}_t) = \frac{1}{n_t} \sum_{i=1}^{n_t} \phi(\mathbf{x}_t + \delta \mathbf{u}_i)\, \mathbf{u}_i \tag{2}$$

where δ is a small positive parameter and $\mathbf{u}_1, \ldots, \mathbf{u}_{n_t}$ are i.i.d. draws from either the uniform distribution over $\mathbb{S}^{d-1}$, the $(d-1)$-dimensional unit sphere, or the multivariate normal distribution. After finding a path, most attacks use variations of binary search to find $\mathbf{x}_{t+1}$. This is generally achieved by finding a boundary point $\mathbf{x}_{t+1}$ along the selected path, where $S(\mathbf{x}_{t+1}) = 0$, using binary search.

As highlighted by [14], the issue with binary search and Eq. (2) is that, in an asymmetric setup where $c^\star > 1$, crafting adversarial examples using these operations becomes costly because approximately half of the generated queries are high-cost. This raises the question of whether we can alter the distribution of generated queries to reduce the number of high-cost queries while maintaining the effectiveness of these two operations as observed in vanilla attacks. As a solution, we propose **AS** and **AGREST** techniques in the following sections.

3.1 Asymmetric Search (AS)

We define $T : [0,1] \to \mathbb{R}^d$ as the function that parameterizes the search path. For example, when the path is a straight line between the source image $\mathbf{x}^\star$ and an adversarial image $\tilde{\mathbf{x}}$ [8, 3], the parameterization is given by $T(\theta) = \theta\,\mathbf{x}^\star + (1-\theta)\,\tilde{\mathbf{x}}$. Similarly, when the search path follows a circular arc [16, 18] between $\mathbf{x}^\star$ and $\tilde{\mathbf{x}}$, lying on a circle in the 2D subspace spanned by $\mathbf{u} = \frac{\tilde{\mathbf{x}}-\mathbf{x}^\star}{\|\tilde{\mathbf{x}}-\mathbf{x}^\star\|_2}$ and a unit vector $\mathbf{v}$ satisfying $\langle \mathbf{u}, \mathbf{v} \rangle = 0$, the parameterization is given by $T(\theta) = \mathbf{x}^\star + \cos\left(\frac{\pi}{2}\theta\right) \cdot \|\tilde{\mathbf{x}} - \mathbf{x}^\star\|_2 \left(\cos\left(\frac{\pi}{2}\theta\right)\mathbf{u} + \sin\left(\frac{\pi}{2}\theta\right)\mathbf{v}\right)$.

As mentioned, the objective is to find the along-the-path boundary point $T(\theta^\star)$ for some $\theta^\star \in [0,1]$. However, since the search space is discrete in practice, our goal is to find a point $\mathbf{x}_{t+1}$ that is near the boundary, i.e., where $S(\mathbf{x}_{t+1}) \approx 0$. This implies that for a given error threshold τ, we need to find $[a,b] \subset [0,1]$ such that $0 < b - a \leq \tau$ and $\theta^\star \in [a,b]$. Since S is continuous, one approach to achieve this is to find $0 \leq k \leq \lceil \frac{1}{\tau} \rceil$ for which $\phi(T(k\tau)) = 1$ and $\phi(T((k+1)\tau)) = -1$. The smaller τ is, the closer we get to the boundary, although this requires more queries.

Algorithm 1 AS Algorithm

Inputs: Parametrization function T, threshold τ
Outputs: Near-boundary adversarial example

$b_l \leftarrow 0,\ b_u \leftarrow \lceil \frac{1}{\tau} \rceil$
while $b_u - b_l > 1$ **do**
  $b_m \leftarrow b_l + \left\lceil \frac{b_u - b_l}{(c^\star + 1)} \right\rceil$
  if $\phi(T(b_m \tau)) = 1$ **then**
    $b_l \leftarrow b_m$
  else
    $b_u \leftarrow b_m$
  end if
end while
return $T(b_l \tau)$

It is well-known that in one-dimensional search, binary search is the optimal comparison-based algorithm in terms of minimizing the expected number of queries, assuming that the boundary point is uniformly distributed along the path (see Assumption A1). Nonetheless, in the asymmetric cost setting, the expected cost of binary search is $\Theta(c^\star \log(1/\tau))$, because the expected number of queries is $\Theta(\log(1/\tau))$, and about half of these queries are expected to incur the higher cost $c^\star$.

Assumption A1. *Let $\Theta^\star_{rv} \in [0,1)$ be a random variable. If $T(\Theta^\star_{rv})$ lies on the decision boundary, that is, $S(T(\Theta^\star_{rv})) = 0$, then we assume $\Theta^\star_{rv}$ is drawn uniformly from $[0,1)$.*

The core idea behind AS is similar to that of binary search, but with a more conservative strategy to account for asymmetric costs. Instead of splitting the interval into two equal parts, AS divides it with a $1 : c^\star$ ratio at each step, favoring lower-cost queries. More specifically, suppose we know the desired point lies within $[b_l\tau, b_u\tau] \subset [0,1]$. Then, as shown in Alg. 1, if $\phi\left(T\left(b_l\tau + \left\lceil \frac{b_u - b_l}{c^\star+1} \right\rceil \tau\right)\right) = 1$, AS continues the search in $\left[b_l\tau + \left\lceil \frac{b_u - b_l}{c^\star+1} \right\rceil \tau, b_u\tau\right]$; otherwise, it proceeds within $\left[b_l\tau, b_l\tau + \left\lceil \frac{b_u - b_l}{c^\star+1} \right\rceil \tau\right]$. This process is repeated until AS locates a point within τ of the boundary.

Note that when $c^\star = 1$, AS reduces to standard binary search, and when $c^\star = \infty$, it becomes a simple line search strategy, as used in stealthy attacks [14], where the algorithm checks $\tau, 2\tau, 3\tau, \ldots$ sequentially. The expected cost of AS is given in Thm. 1, showing that it improves over binary search by a factor of $\Theta(\log(c^\star + 1))$.

Theorem 1. ***(Cost Analysis of AS)*** *Suppose $0 < \tau < 1$ and $c^\star \geq 1$. Under Assumption A1, the expected cost of the AS algorithm is $\mathcal{O}(c^\star \log_{(c^\star+1)}(1/\tau))$.*

To illustrate the effect of AS in practice, we compare the cost of AS and binary search when $c^\star = 10^3$, and we observe that the cost of binary search is approximately 2.5 times higher than that of AS. The results are provided in App. C. An illustration of the AS algorithm can be found in App. H (Right) and Fig. 8.

3.2 Asymmetric Gradient Estimation (AGREST)

As mentioned earlier, to keep the gradient estimation (i.e., Eq. (2)) effective in the asymmetric setup, our goal is to adjust the distribution of queries generated during the process. To achieve this, we first propose a family of estimations that provides flexibility in adjusting the distribution of queries used for estimation. Then, we introduce a method to select the estimator within this family that maximizes the similarity between the approximated gradient and the true gradient.

How can we control the query distribution? The main idea behind AGREST is to estimate $\nabla S(\mathbf{x}_t)$ by approximating the gradient at a point like $\mathbf{x}'_t$, which is close to $\mathbf{x}_t$. This approach allows us to alter the distribution of made queries while approximating effective directions for the attacks. In this method, we move $\mathbf{x}_t$ away from $\mathbf{x}^\star$ by ω_t, the overshooting value, to reach the new point $\mathbf{x}'_t = \mathbf{x}_t + \omega_t \frac{\mathbf{x}_t - \mathbf{x}^\star}{\|\mathbf{x}_t - \mathbf{x}^\star\|_2}$(Fig. 2 (left)). Then, we use Eq. (2) at $\mathbf{x}'_t$ almost similar to the vanilla estimation, except that AGREST assigns more weight to high-cost queries than to low-cost ones (using importance sampling). In other words, we estimate ∇S as follows:

$$\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t) = \frac{1}{n_t}\sum_{i=1}^{n_t} \widehat{\phi}_t(\mathbf{x}'_t + \delta \mathbf{u}_i)\,\mathbf{u}_i = \frac{1}{n_t}\sum_{i=1}^{n_t} \widehat{\phi}_t\left(\mathbf{x}_t + \omega_t \frac{\mathbf{x}_t - \mathbf{x}^\star}{\|\mathbf{x}_t - \mathbf{x}^\star\|_2} + \delta \mathbf{u}_i\right)\mathbf{u}_i$$

where $\widehat{\phi}_t(\mathbf{x}) = (1-\beta_t)\,\mathbf{1}\{\phi(\mathbf{x}) = 1\} - \beta_t\,\mathbf{1}\{\phi(\mathbf{x}) = -1\}$ is the sampling weight function, $\frac{1}{2} \leq \beta_t < 1$ is the sampling weight parameter, $\mathbf{u}_1, \ldots, \mathbf{u}_{n_t}$ are i.i.d. draws from $\textsc{Uniform}(\mathbb{S}^{d-1})$, and $\mathbf{1}\{\cdot\}$ denotes the indicator function. The parameters ω_t and β_t allow us to control the likelihood of making high-cost queries and their associated weight in our estimation.

How can we choose the best AGREST estimator? To ensure the selected estimation is as close as possible to the true gradient direction among all AGREST estimators, one potential solution is to find the estimator that maximizes:

$$\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) = \mathbb{E}_{\mathbf{u}_{1:n_t}} \cos\left(\nabla S(\mathbf{x}_t), \widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right)$$

within the query budget, where cos represents the cosine similarity function. To calculate this function, we need to assume that S has certain characteristics. One common choice is to assume that S is L-smooth. However, this assumption introduces excessive complexity to our analysis and may add additional hyperparameters related to L to the current set of hyperparameters in the existing attacks. Therefore, similar to [8, 18] and based on observations from [19, 20, 21], we assume that S is locally linear around $\mathbf{x}_t$, $S(\mathbf{x}'_t + \delta\mathbf{u}) \approx S(\mathbf{x}_t) + \langle \nabla S(\mathbf{x}_t), \mathbf{x}'_t + \delta \mathbf{u} - \mathbf{x}_t\rangle$. Since $\mathbf{x}_t$ is a boundary point, $S(\mathbf{x}_t) = 0$. Thus, we have:

$$\phi(\mathbf{x}'_t + \delta\mathbf{u}) \approx \operatorname{sign}\left(\langle \nabla S(\mathbf{x}_t), \mathbf{x}'_t + \delta\mathbf{u} - \mathbf{x}_t\rangle\right) = \operatorname{sign}(\cos\alpha_t \cdot \omega_t + \langle \mathbf{g}_t, \delta\mathbf{u}\rangle) \tag{3}$$

where $\mathbf{g}_t = \frac{\nabla S(\mathbf{x}_t)}{\|\nabla S(\mathbf{x}_t)\|_2}$ and α_t is the angle between $\mathbf{x}_t - \mathbf{x}^\star$ and $\mathbf{g}_t$ (Fig. 2). Additionally, based on this assumption, we can calculate the probability of low-cost queries, namely $p_t(\omega_t) = \mathbb{P}[\phi(\mathbf{x}'_t + \delta\mathbf{u}) = 1]$, using Lem. 1.

Lemma 1. ***(Hyperspherical Cap [22])*** *Under local linearity around $\mathbf{x}_t$ (Eq. (3)), we have $p_t(\omega_t) = \frac{1}{2}\left(1 + \mathcal{I}_{d-2}(\delta^{-1}\cos\alpha_t\omega_t)/\mathcal{I}_{d-2}(0)\right)$, where $\mathcal{I}_d(x) = \int_0^{1-x}(1-t^2)^{(d-1)/2}dt$.*

Based on Lem. 1, we can infer that $p_t(\omega_t)$ is strictly increasing and therefore invertible. Nonetheless, even with the linearity assumption, calculating the expected value remains challenging due to the nonlinearity of cosine similarity and the complexity of handling multiple independent random vectors. Therefore, inspired by measure concentration [23], we approximate $\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t)$ as follows:

$$J(\mathbf{x}_t, \omega_t, \beta_t, n_t) = \left(n_t^{1/2} \cdot \mathbb{E}\left[\widehat{\phi}(\mathbf{x}'_t + \delta\mathbf{u})\langle \mathbf{g}_t, \mathbf{u}\rangle\right]\right) \cdot \left(\mathbb{E}\left[\widehat{\phi}(\mathbf{x}'_t + \delta\mathbf{u})^2\right]\right)^{-1/2}$$

This new objective is easier to calculate since it removes the need to deal with multiple random vectors. Thm. 2 establishes a convergence bound for the approximation.

Theorem 2. ***(Expected Cosine Similarity Approximation)*** *Under the local linearity assumption around $\mathbf{x}_t$, for any constants $0 < z < \frac{1}{8}$ and $\frac{1}{2} \leq q, \beta < 1$, as n_t and d approach infinity, we have*

$$\left|\frac{\mu(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} - 1\right| \leq \mathcal{O}\left(d^{-z}\right)$$

Now, we can formulate the optimization problem. The goal is to maximize $J(\mathbf{x}_t, \omega_t, \beta_t, n_t)$ within a query budget. Specifically, we want to:

$$\max_{\omega_t, \beta_t, n_t} \quad J(\mathbf{x}_t, \omega_t, \beta_t, n_t) \qquad \text{s.t.} \quad n_t(c^\star - (c^\star - 1)p_t(\omega_t)) \leq c_t \tag{4}$$

where c_t is the maximum allowed cost of estimation at iteration t of the algorithm. The constraint in Eq. (4) ensures that the expected estimation cost at iteration t is at most c_t. To solve this optimization problem, we propose Thm. 3.

Theorem 3. ***(Optimal AGREST Parameters)*** *Suppose the solution to Eq. (4) is represented by* $(\omega_t^\star, w_t^\star, n_t^\star)$*. Given the local linearity around* $\mathbf{x}_t$*, the following statements hold:*

1. $n_t^\star = c_t \left(c^\star - (c^\star - 1)p_t(\omega_t^\star)\right)^{-1}$ *and* $\beta_t^\star = p_t(\omega_t^\star)$.
2. $\omega_t^\star$ *maximizes the following function over the interval* $[0, \delta/\cos\alpha_t]$*:*

$$\widehat{J}_t(\omega_t) = \left(1 - \left(\delta^{-1}\cos\alpha_t\omega_t\right)^2\right)^{d-1}\left(p_t(\omega_t)\left(1 - p_t(\omega_t)\right)\left(c^\star - (c^\star - 1)p_t(\omega_t)\right)\right)^{-1}$$

An immediate consequence of Lem. 1 and Thm. 3 is the existence of $\omega^\star$ such that $\omega^\star = \cos\alpha_1 \cdot \omega_1^\star = \cos\alpha_2 \cdot \omega_2^\star = \ldots$. As a result, we aim to find $\omega^\star$. One problem is that $\mathcal{I}_d$ has a complex closed form. Thus, finding a closed form for $\omega^\star$ would be challenging. Instead, we use numerical methods to evaluate this integral and numerical optimization techniques to find $\omega^\star$. In particular, we use QUADPACK [24] for the integral calculation and the Nelder–Mead method [25] for maximizing $\widehat{J}_t(\omega_t)$. Another problem is that we need to know α_t in order to obtain the optimal overshooting value, which is not possible in a black-box setup. Thus, we need to estimate α_t at each iteration t.

How can we estimate α_t? We split this problem into two steps: 1. estimating α_1, and 2. understanding the behavior of α_t with respect to t. For the first problem, initially, we expect $\mathbf{x}_1 - \mathbf{x}^\star$ and $\mathbf{g}_1$ to be somehow independent, as most existing attacks select $\mathbf{x}_1$ using a random direction. The only reasonable assumption about these two vectors is that they likely have a positive correlation, i.e., $\langle \mathbf{x}_1 - \mathbf{x}^\star, \mathbf{g}_1\rangle \geq 0$. Specifically, if we know that $\mathbf{x}_1$ is the closest boundary point to $\mathbf{x}^\star$ along the direction of $\mathbf{x}_1 - \mathbf{x}^\star$, meaning there is no $0 < r < 1$ such that $\phi(\mathbf{x}^\star + r(\mathbf{x}_1 - \mathbf{x}^\star)) = 1$, this assumption provably holds. Given this assumption, we can use Thm. 4 to attain α_1.

Algorithm 2 Overshooting Scheduler Step

Inputs: Iteration t, dimension d, desired probability p, scheduler rate m
Outputs: Next cosine value α_{t+1}

$\alpha_1 \leftarrow$ INIT-ANGLE (d) ▷ Thm. 4
$\hat{\alpha}_{t+1} \leftarrow 1 - (1 - \cos\alpha_1)(t+1)^{-m}$
$\alpha_{t+1} \leftarrow \arccos(\hat{\alpha}_{t+1})$
return α_{t+1}

Theorem 4. ***(Initial Cosine Value)*** *Under local linearity around* $\mathbf{x}_1$*, if there is no* $0 < r < 1$ *such that* $\phi(\mathbf{x}^\star + r(\mathbf{x}_1 - \mathbf{x}^\star)) = 1$*, then we have* $\mathbb{E}[\cos\alpha_1] = \Gamma\left(\frac{d}{2}\right)\left(2\sqrt{\pi}\,\Gamma\left(\frac{d+1}{2}\right)\right)^{-1}$.

Algorithm 3 AGREST Estimation

Inputs: Iteration t, source image $\mathbf{x}^\star$, boundary point $\mathbf{x}_t$, dimension d, high-cost query cost $c^\star$, sampling radius δ, sampling batch size b, cosine value α_t, vanilla gradient estimation query budget n_t', scheduler rate m
Outputs: Normalized approximated direction g_t, next cosine value α_{t+1}

$n_L \leftarrow 0,\ n_H \leftarrow 0,\ \mathbf{v}^+ \leftarrow \vec{0},\ \mathbf{v}^- \leftarrow \vec{0},\ \hat{c} \leftarrow 0,\ \omega^\star \leftarrow$ OVERSHOOTING $(c^\star)$ ▷ Thm. 3
$\omega_t \leftarrow \omega^\star/\cos\alpha_t,\ c_t \leftarrow n_t'(c^\star + 1)/2$
while $\hat{c} < c_t$ **do**
  for each $\mathbf{u}_i \sim$ UNIFORM$(\mathbb{S}^{d-1})$, $i = 1, \ldots, b$ **do**
    if $\phi\left(\mathbf{x}_t + \omega_t \frac{\mathbf{x}_t - \mathbf{x}^\star}{\|\mathbf{x}_t - \mathbf{x}^\star\|_2} + \delta\mathbf{u}_i\right) = 1$ **then**
      $\mathbf{v}^+ \leftarrow \mathbf{v}^+ + \mathbf{u}_i,\ n_L \leftarrow n_L + 1,\ \hat{c} \leftarrow \hat{c} + 1$
    else
      $\mathbf{v}^- \leftarrow \mathbf{v}^- - \mathbf{u}_i,\ n_H \leftarrow n_H + 1,\ \hat{c} \leftarrow \hat{c} + c^\star$
    end if
  end for
end while
$\hat{p}_t \leftarrow n_L/(n_L + n_H)$
$g_t \leftarrow (1 - \hat{p}_t)\mathbf{v}^+ + \hat{p}_t\mathbf{v}^-,\ \alpha_{t+1} \leftarrow$ SCHEDULER-STEP $(t, \hat{p}_t, m)$ ▷ Alg. 2
return $g_t/\|g_t\|_2,\ \alpha_{t+1}$

The next step is to estimate α_t after the first iteration. Chen et al. [3] showed that in HSJA, $\cos(\mathbf{x}_t - \mathbf{x}^\star, \mathbf{g}_t) \geq 1 - c\,t^{-m}$ for some constant c and $0 < m < \frac{1}{2}$. This motivated us to heuristically estimate α_t as $\arccos(1 - (1 - \cos\alpha_1)t^{-m})$, where m is a new hyperparameter introduced to the existing

Table 1: Median ℓ_2 distance for various $c^\star$ values and different types of attacks across neural network architectures. VA stands for Vanilla Attack. The bold numbers represent the best performance among different variants of each attack for each $c^\star$ value and model (For a comprehensive analysis of attacks under varying total cost constraints, we refer readers to Tab. 2 and Tab. 3 in App. E, which present exhaustive experimental results across different total cost budgets and query cost $c^\star$.)

		ResNet-50				ViT-B/32			
		$c^\star = 2$	$c^\star = 5$	$c^\star = 10^2$	$c^\star = 10^3$	$c^\star = 2$	$c^\star = 5$	$c^\star = 10^2$	$c^\star = 10^3$
Attack	*Method*	*Total Cost of Queries*							
		10K	15K	150K	250K	10K	15K	150K	250K
SurFree	VA	4.09	5.19	5.21	17.49	2.9	2.5	5.13	16.12
	VA+AS (A-SurFree)	**3.45**	**3.52**	**3.80**	**7.59**	**2.4**	**2.1**	**3.68**	**6.35**
HSJA	VA	2.24	2.77	4.66	23.72	18.3	13.9	4.21	22.46
	VA+AS	2.16	2.72	4.09	19.07	18.3	13.5	3.88	18.79
	VA+AGREST	2.19	2.51	2.49	14.62	2.7	2.2	2.19	11.28
	VA+AS+AGREST (A-HSJA)	**2.13**	**2.39**	**2.16**	**12.28**	**2.7**	**2.1**	**2.06**	**10.74**
GeoDA	VA	2.80	3.21	4.02	10.80	2.7	2.4	3.97	9.83
	VA+AS	**2.66**	3.12	3.32	9.24	2.7	2.3	3.12	8.78
	VA+AGREST	2.89	2.95	2.19	6.28	**1.9**	**1.7**	2.10	5.12
	VA+AS+AGREST (A-GeoDA)	2.93	**2.8**	**2.11**	**5.78**	1.9	1.8	**2.03**	**4.35**
CGBA	VA	1.21	1.42	2.22	9.97	1.6	1.4	2.13	9.67
	VA+AS	1.17	1.39	2.06	9.28	1.7	1.3	1.97	8.24
	VA+AGREST	**1.12**	1.36	1.63	**5.73**	1.5	1.3	1.56	5.46
	VA+AS+AGREST (A-CGBA)	1.15	**1.33**	**1.58**	6.23	**1.5**	**1.2**	**1.42**	**5.61**

attacks (Alg. 2). As stated in Thm. 3, under the assumption of local linearity, the value of β_t in an optimal AGREST estimation is the probability of making low-cost queries. However, similar to [3], we use the empirical probability of making low-cost queries, namely $\frac{n_{\mathrm{L}}}{n_{\mathrm{L}}+n_{\mathrm{H}}}$, to reduce the variance of the estimation. Here, n_{H} and n_{L} represent the number of high-cost and low-cost queries made in an AGREST estimator, respectively. Additionally, we set c_t to the expected cost of the vanilla attack, namely $\frac{n'_t(c^\star+1)}{2}$, where n'_t is the number of made queries by the vanilla attack at iteration t. A detailed overview of AGREST is provided in Alg. 3. Note that in practice, most attacks are performed in a given subspace rather than in the entire space to improve sample efficiency. In these cases, we use the effective dimension d' of the subspace instead of d, the dimension of the original space. For more details, see App. D. Finally, it is worth mentioning that the probability of making low-cost queries in AGREST closely follows our theoretical analysis in practice. Further details and empirical results are presented in App. C.

4 Experiments

Model, dataset, and metric. For robust evaluation of asymmetric attacks, we employed ImageNet-trained models: ResNet-50 [26], ViT-B/32, ViT-B/16 [27], and CLIP [28]. Original images ($\mathbf{x}^\star$) were 500 correctly classified ImageNet validation samples similar to the [14]. Numerical tasks used SciPy [29]. Attack performance was measured by the median ℓ_2 distance between perturbations and originals over query costs, consistent with previous work.

Attacks and hyperparameters. To evaluate our proposed framework, we modify SurFree, HSJA, GeoDA, and CGBA by changing their respective search methods to AS and, where applicable, updating their gradient estimation to asymmetric gradient estimation. The other components of the attacks remain similar to their original implementations, except for the line search along the gradient direction in GeoDA. We slightly modify this part to reduce the number of high-cost queries generated during this phase (see App. D). Moreover, to compare our framework with stealthy attacks, we use Stealthy HSJA, as it outperforms other stealthy attacks when the ℓ_2 norm is the evaluation metric (Fig. 3 of [14]). For the hyperparameters used in the vanilla attacks, we generally use the same values. The only exception is the subspace method in SurFree. Specifically, instead of using the $\mathrm{DCT}_{8\times8}$ method in SurFree, we set it to $\mathrm{DCT}_{\mathrm{full}}$. This adjustment allows for a fair comparison of SurFree with other attacks, as GeoDA and CGBA both use the $\mathrm{DCT}_{\mathrm{full}}$ technique. Furthermore, we set the newly introduced hyperparameter m, the overshooting scheduler rate, to 0.02, 0.06, and 0.06 for HSJA, GeoDA, and CGBA, respectively (for more details on the effect of hyperparameter m see App. D).

Ablation study. To evaluate the effectiveness of AS and AGREST in different attacks, we test various combinations of these two approaches with each attack when $c^\star = 2$, 5, 10^2, or 10^3 (Tab. 1). As shown in Tab. 1, for SurFree, we compare the vanilla attack with **A-SurFree**, the new asymmetric attack that utilizes AS. As expected, replacing binary search with AS leads to smaller adversarial perturbations for all $c^\star$. Furthermore, we compare the performance of gradient-based attacks with their corresponding variations, namely: 1. Replacing binary search with AS 2. Replacing vanilla gradient estimation with AGREST 3. Combining the two previous approaches to obtain **A-HSJA**, **A-GeoDA**, **A-CGBA**. In general, using asymmetric attacks results in incremental improvements when $c^\star = 2$ or 5. This is expected because we anticipate binary search and vanilla gradient estimation perform well when the cost of high-cost queries is not significantly different from the cost of low-cost queries. However, for larger values of $c^\star$, namely 10^2 and 10^3, the improvements are substantial. Specifically, using AGREST alone reduces the ℓ_2 norm by approximately 40% in all cases. Moreover, combining AGREST with AS further decreases the norm. One notable point is that AGREST enhances attacks utilizing gradient estimation more than AS does. This occurs because attacks using gradient estimation spend most of their query budget on gradient approximation rather than on search. (Tab. III of [14])

Comparison to *stealthy* attacks. As mentioned, for larger values of $c^\star$, we expect asymmetric attacks to significantly improve over vanilla attacks. Nonetheless, in these cases, we must compare our framework to stealthy attacks, since, unlike with lower to medium values of $c^\star$, stealthy attacks outperform vanilla attacks when $c^\star$ is large (Fig. 7 of [14]). As a result, we evaluate the performance of A-SurFree, A-HSJA, A-GeoDA, and A-CGBA against Stealthy HSJA on the ResNet model when $c^\star = 10^4$, 10^5, or ∞. As demonstrated in Fig. 3, when $c^\star = 10^4$, all asymmetric attacks, including A-HSJA which retains the gradient estimation method that Stealthy HSJA discards, outperform Stealthy HSJA. The same holds for $c^\star = 10^5$ (Fig. 3). For the case where $c^\star = \infty$, following [14], we determine the cost of each attack by counting the number of high-cost (flagged) queries it generates. In this setup, we assume $c^\star = 10^5$ during the execution of AGREST and AS. As shown in Fig. 3, all asymmetric attacks outperform Stealthy HSJA by a wide margin.

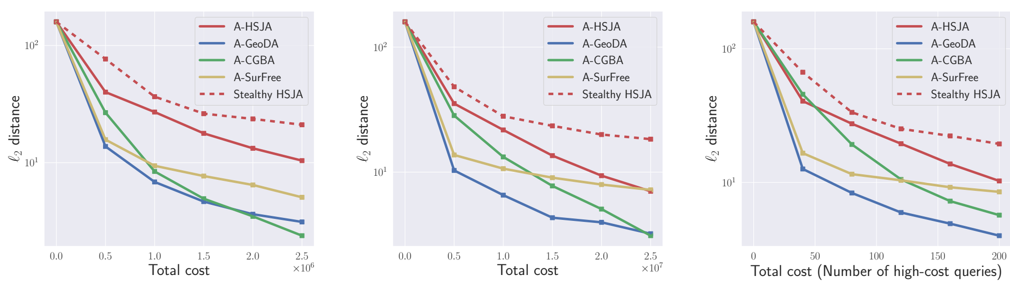

Figure 3: **Performance of various asymmetric attacks compared to Stealthy HSJA under high cost asymmetry with ResNet-50.** The value of $c^\star$ is 10^4, 10^5, and ∞ from left to right.

Asymmetric attacks against CLIP. We evaluate CLIP [28] as a representative vision-language model (VLM) under both zero-shot and fine-tuned settings. Our asymmetric attack achieves significantly better performance than stealthy baselines; results are provided in App. G.

5 Conclusion and outlook

We proposed a framework that extends existing decision-based black-box attacks to handle asymmetric query costs, where querying the source class is more expensive than others. Our method modifies two core components, gradient estimation and search, and achieves significant improvements over both standard and stealthy attack baselines. However, it introduces a new hyperparameter, which may require tuning for different settings. Additionally, we assume local linearity around decision boundaries; while this assumption is common in the adversarial examples literature, it may not hold for some models, potentially affecting the accuracy of gradient estimates. There are also many interesting directions for future work, such as generalizing the framework beyond the binary setting of source versus non-source classes. For instance, different target classes may each have their own associated query cost, or the cost may depend on the classifier's confidence, as in score-based scenarios. Applying our framework to vision-language models such as Vision LLaMA [30, 31] or LLaVA is another promising direction. AS could also enhance jailbreak attacks on large language models, potentially replacing random search-based methods [32, 33], though adapting our framework to LLMs presents challenges due to the discrete nature of text prompts [34]. Exploring these avenues could expand the impact of asymmetric attacks across a wide range of applications.

References

[1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

[2] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[3] 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.

[4] Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack, 2020.

[5] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018.

[6] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

[7] Yujia Liu, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack, 2019.

[8] Ali Rahmati, Seyed Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: a geometric framework for black-box adversarial attacks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8446–8455, 2020.

[9] Facebook. How does facebook use artificial intelligence to moderate content? URL `https://www.facebook.com/help/1584908458516247`, 2024. Accessed 03-05-2024.

[10] Twitter. X's sensitive media policy. URL `https://help.twitter.com/en/rules-and-policies/media-policy`, 2024. Accessed 03-05-2024.

[11] Google. Detect explicit content (safe search). URL `https://cloud.google.com/vision/docs/detecting-safe-search`, 2024. Accessed 03-05-2024.

[12] Amazon. Moderating content. URL `https://docs.aws.amazon.com/rekognition/latest/dg/moderation.html`, 2024. Accessed 03-05-2024.

[13] Microsoft. Detect adult, racy, or gory content. URL `https://learn.microsoft.com/azure/ai-services/computer-vision/concept-detecting-adult-content`, 2024. Accessed 03-05-2024.

[14] Edoardo Debenedetti, Nicholas Carlini, and Florian Tramèr. Evading black-box classifiers without breaking eggs. In *2024 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 408–424. IEEE, 2024.

[15] Gabriel Alves, Geoff Pilling, James Innes, Refath Bari, Nguyen Quang, Josh Silverman, Arron Kau, and Jimin Khim. Egg dropping. URL `https://brilliant.org/wiki/egg-dropping/`, 2024. Accessed 26-04-2024.

[16] Md Farhamdur Reza, Ali Rahmati, Tianfu Wu, and Huaiyu Dai. Cgba: Curvature-aware geometric black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 124–133, 2023.

[17] Twitter. Understanding x limits. URL `https://help.twitter.com/en/rules-and-policies/x-limits`, 2024. Accessed 04-05-2024.

[18] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10430–10439, 2021.

[19] Alhussein Fawzi, Seyed Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. *Advances in neural information processing systems*, 29, 2016.

[20] Alhussein Fawzi, Seyed Mohsen Moosavi-Dezfooli, and Pascal Frossard. The robustness of deep networks: A geometrical perspective. *IEEE Signal Processing Magazine*, 34(6):50–62, 2017.

[21] Alhussein Fawzi, Seyed Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2018.

[22] Alexander Mikhailovich Chudnov. On minimax signal generation and reception algorithms. *Problemy Peredachi Informatsii*, 22(4):49–54, 1986.

[23] Michel Ledoux. *The concentration of measure phenomenon*. American Mathematical Soc., 2001.

[24] Robert Piessens, Elise de Doncker-Kapenga, Christoph W Überhuber, and David K Kahaner. *Quadpack: a subroutine package for automatic integration*, volume 1. Springer Science & Business Media, 2012.

[25] John A Nelder and Roger Mead. A simplex method for function minimization. *The computer journal*, 7(4):308–313, 1965.

[26] 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.

[27] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[28] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

[29] Pauli Virtanen, Ralf Gommers, Travis E Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, et al. Scipy 1.0: fundamental algorithms for scientific computing in python. *Nature methods*, 17(3):261–272, 2020.

[30] Xiangxiang Chu, Jianlin Su, Bo Zhang, and Chunhua Shen. Visionllama: A unified llama backbone for vision tasks, 2024.

[31] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.

[32] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks. *arXiv preprint arXiv:2404.02151*, 2024.

[33] Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.

[34] Elias Abad Rocamora, Grigorios G. Chrysos, and Volkan Cevher. Certified robustness under bounded levenshtein distance, 2025.

[35] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[36] 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.

[37] 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.

[38] Alireza Abdolahpourrostam, Mahed Abroshan, and Seyed-Mohsen Moosavi-Dezfooli. Superdeepfool: a new fast and accurate minimal adversarial attack. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

[39] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.

[40] 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, 2017.

[41] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International conference on machine learning*, pages 2137–2146. PMLR, 2018.

[42] Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

[43] Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566, 2017.

[44] Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. Qeba: Query-efficient boundary-based blackbox attack. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1221–1230, 2020.

[45] Yujia Liu, Seyed Mohsen Moosavi-Dezfooli, and Pascal Frossard. A geometry-inspired decision-based attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4890–4898, 2019.

[46] Xiaosen Wang, Zeliang Zhang, Kangheng Tong, Dihong Gong, Kun He, Zhifeng Li, and Wei Liu. Triangle attack: A query-efficient decision-based adversarial attack. In *European conference on computer vision*, pages 156–174. Springer, 2022.

[47] Vitali D Milman and Gideon Schechtman. *Asymptotic theory of finite dimensional normed spaces: Isoperimetric inequalities in riemannian manifolds*, volume 1200. Springer Science & Business Media, 1986.

[48] Herman Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, pages 493–507, 1952.

[49] Mor Harchol-Balter. *Introduction to probability for computing*. Cambridge University Press, 2023.

A Related work

Decision-based attacks. Adversarial examples can be crafted in three setups: white-box [35, 36, 37, 38], score-based black-box [39, 40, 41], and decision-based black-box [1]. In the decision-based black-box setup, the attacker relies solely on predictions without access to models or class scores. Moreover, decision-based attacks can be either targeted or non-targeted. Non-targeted attacks craft adversarial examples without any constraints on the model's prediction for the adversarial example. Boundary Attack [1] was the first effective decision-based attack based on random walking. OPT [5] outperforms Boundary Attack by introducing a gradient-based approach. Inspired by zeroth-order optimization methods [42, 43], HSJA estimates the gradient of the classification margin without direct access to the margin itself and minimizes perturbation size using optimization techniques [3]. QEBA [44] uses various techniques to approximate the gradient of the classification margin more effectively than HSJA, leveraging insights like local similarity and the importance of the low-frequency subspace. GeoDA and qFool [8, 45] use techniques similar to HSJA's gradient estimation to locally approximate the decision boundary as a hyperplane at each iteration. They then search for optimal adversarial examples based on these estimated hyperplanes. While gradient-based methods outperform previous approaches, their reliance on generating numerous queries for efficient gradient estimation led [18] to focus on decision boundary geometry. They introduced SurFree, which iteratively selects a random 2D subspace and searches for adversarial examples along a circular path. In a similar way, TriA [46] generates effective adversarial examples while using minimal queries. CGBA [16] combines the gradient approximation method from GeoDA with SurFree's 2D subspace search technique to achieve state-of-the-art results.

Asymmetric query costs. Existing decision-based attacks assume all queries have the same cost. However, [14] showed this may not be the case in real-world scenarios. They found that queries belonging to the target class can be problematic in certain situations and noted that all decision-based attacks produce many of these bad queries. To mitigate this, they introduced stealthy attacks inspired by the egg dropping problem [15]. However, stealthy attacks face two main challenges. First, they overlook the cost of queries that are not bad. For example, their most effective attack, Stealthy HSJA, generates about 10^7 queries for every 1,000 bad queries. Second, to reduce the number of bad queries during the gradient estimation phase, [14] replaced the HSJA gradient approximation, known for its benefits in crafting adversarial examples, with OPT gradient estimation, believing that modifying the HSJA gradient estimation to perform well in this new setup would be difficult. In this paper, we address these challenges in non-targeted decision-based attacks. In particular, we find a way to efficiently distribute our total query budget between problematic and non-problematic queries while keeping the HSJA method of gradient approximation by slightly modifying the method.

B Proofs

Proof of Thm. 1 We define the expected cost of the algorithm for $1 < m < \lceil \frac{1}{\tau} \rceil$ points as $C(m)$. Using Alg. 1 and applying the law of total expectation, we obtain

$$C(m) = \mathbb{P}\left[\phi\left(T\left(b_m\tau\right)\right) = -1\right]\left(C\left(\frac{m}{c^\star+1}\right)+c^\star\right)+\mathbb{P}\left[\phi\left(T\left(b_m\tau\right)\right) = 1\right]\left(C\left(\frac{c^\star m}{c^\star+1}\right)+1\right). \tag{5}$$

We now claim that

$$C(m) < 2c^\star \left\lceil \log_{c^\star+1} m \right\rceil. \tag{6}$$

We prove this by induction.

For the base case, when $m \leq c^\star + 1$, Alg. 1 reduces to a simple line search. Assuming a uniform distribution for the boundary point, the expected cost of the line search is approximately $c^\star + m/2$, which clearly satisfies Eq. (6).

For the induction step, suppose the claim holds for all values smaller than m. Under the uniform distribution assumption for the boundary point, we have

$$\mathbb{P}\left[\phi\left(T\left(b_m\tau\right)\right) = -1\right] = \frac{1}{c^\star+1}, \quad \mathbb{P}\left[\phi\left(T\left(b_m\tau\right)\right) = 1\right] = \frac{c^\star}{c^\star+1}.$$

Substituting into Eq. (5) and applying the induction hypothesis yields:

$$\begin{aligned} C(m) &= \frac{1}{c^\star+1}\left(C\left(\frac{m}{c^\star+1}\right)+c^\star\right)+\frac{c^\star}{c^\star+1}\left(C\left(\frac{c^\star m}{c^\star+1}\right)+1\right) \\ &< \frac{1}{c^\star+1}\left(2c^\star\left\lceil\log_{c^\star+1}\frac{m}{c^\star+1}\right\rceil+c^\star\right)+\frac{c^\star}{c^\star+1}\left(2c^\star\left\lceil\log_{c^\star+1}\frac{c^\star m}{c^\star+1}\right\rceil+1\right). \end{aligned}$$

Noting that

$$\log_{c^\star+1}\left(\frac{m}{c^\star+1}\right) = \log_{c^\star+1}(m)-1, \quad \log_{c^\star+1}\left(\frac{c^\star m}{c^\star+1}\right) = \log_{c^\star+1}(m)+\log_{c^\star+1}\left(\frac{c^\star}{c^\star+1}\right),$$

and observing that $\log_{c^\star+1}\left(\frac{c^\star}{c^\star+1}\right) < 0$ but close to 0 for large $c^\star$, we can bound both ceilings by $\lceil \log_{c^\star+1}(m) \rceil$.

Thus,

$$\begin{aligned} C(m) &< \frac{2c^\star}{c^\star+1}\left(\lceil\log_{c^\star+1} m\rceil-1\right)+\frac{c^\star}{c^\star+1}c^\star+\frac{2{c^\star}^2}{c^\star+1}\lceil\log_{c^\star+1} m\rceil+\frac{c^\star}{c^\star+1} \\ &= 2c^\star\lceil\log_{c^\star+1} m\rceil. \end{aligned}$$

Thus, the expected complexity of the algorithm is

$$\mathcal{O}\left(\frac{c^\star\log\left(1/\tau\right)}{\log(c^\star+1)}\right),$$

completing the proof.

■

Proof of Thm. 2 Before proving the theorem, we first introduce some useful lemmas.

Lemma 2. ***(Lévy's Lemma [47, 23])*** *Let $f : \mathbb{S}^{d-1} \to \mathbb{R}$ be an L-Lipschitz function on the unit hypersphere, and let $\mathbf{x} \sim$ Uniform $\left(\mathbb{S}^{d-1}\right)$. Then, for some constant $C > 0$, we have:*

$$\mathbb{P}\left(\left|f\left(\mathbf{x}\right)-\mathbb{E}f\left(\mathbf{x}\right)\right| > \varepsilon\right) \leq 2\exp\left(-\frac{Cd\varepsilon^2}{L^2}\right).$$

Corollary 1. *Suppose $\mathbf{u} \in \mathbb{R}^d$ is a unit vector, and $\mathbf{x} \sim$ Uniform $\left(\mathbb{S}^{d-1}\right)$. Then, for some constant $C > 0$, we have:*

$$\mathbb{P}\left(\left|\langle\mathbf{u},\mathbf{x}\rangle\right| > \varepsilon\right) \leq 2\exp\left(-Cd\varepsilon^2\right).$$

Lemma 3. *Suppose $\widehat{\phi}^2_{t,1}, \ldots, \widehat{\phi}^2_{t,n_t}$ are i.i.d. Bernoulli random variables with support $\{\beta_t^2, (1-\beta_t)^2\}$, where $\beta_t > \frac{1}{2}$. If $\widehat{\phi}_t$ is an i.i.d. copy of $\widehat{\phi}_{t,i}$, then the following inequality holds:*

$$\mathbb{P}\left(\left|\frac{1}{n_t}\sum_{i=1}^{n_t}\widehat{\phi}^2_{t,i}-\mathbb{E}[\widehat{\phi}^2_t]\right| > \varepsilon\right) \leq 2\exp\left(-\frac{2n_t\varepsilon^2}{(2\beta_t-1)^2}\right).$$

Proof of Lem. 3: The result follows by applying the Chernoff bound for binomial distributions [48, 49] to the transformed random variables

$$\widetilde{\phi}_i := \frac{\widehat{\phi}_{t,i}^2 - (1-\beta_t)^2}{\beta_t^2 - (1-\beta_t)^2},$$

which are i.i.d. Bernoulli random variables taking values in $\{0,1\}$.

Lemma 4. *Given the local linearity around $\mathbf{x}_t$, for any $\omega_t \in \left[0, \frac{\delta}{\cos\alpha_t}\right]$, we have:*

$$\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right] = \frac{\Gamma\left(\frac{d}{2}\right)\cdot\left(1-(\cos\alpha_t\cdot\omega_t/\delta)^2\right)^{\frac{d-1}{2}}}{2\sqrt{\pi}\cdot\Gamma\left(\frac{d+1}{2}\right)}$$

Proof of Lem. 4: By the law of total expectation and linearity assumption, we have

$$\begin{aligned}\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right] &= \beta_t\left(1-P_t(\omega_t)\right)\mathbb{E}\left[-\langle \mathbf{g}_t, \mathbf{u}\rangle | \langle \mathbf{g}_t, \mathbf{u}\rangle \le -\cos\alpha_t\cdot\omega_t/\delta\right] \\ &\quad + (1-\beta_t)\,P_t(\omega_t)\,\mathbb{E}\left[\langle \mathbf{g}_t, \mathbf{u}\rangle | \langle \mathbf{g}_t, \mathbf{u}\rangle > -\cos\alpha_t\cdot\omega_t/\delta\right]\end{aligned} \tag{7}$$

Now, by applying the divergence theorem on the constant vector field $\vec{\mathbf{F}} = \mathbf{g}_t$, we have

$$\begin{aligned}\mathbb{E}\left[-\langle \mathbf{g}_t, \mathbf{u}\rangle | \langle \mathbf{g}_t, \mathbf{u}\rangle \le -\cos\alpha_t\cdot\omega_t/\delta\right] &= \langle -\mathbf{g}_t, \int_{\langle \mathbf{g}_t, \mathbf{u}\rangle \le -\cos\alpha_t\cdot\omega_t/\delta} \mathbf{u}p(\mathbf{u})\,d\mathbf{u}\rangle \\ &= \langle -\mathbf{g}_t, \int_{\langle \mathbf{g}_t, \mathbf{u}\rangle \le -\cos\alpha_t\cdot\omega_t/\delta} \frac{1}{\left(1-p(\omega_t)\right)A_{d-1}(1)}\mathbf{u}d\mathbf{u}\rangle \\ &= \frac{V_{d-1}\left(\sqrt{1-(\cos\alpha_t\cdot\omega_t/\delta)^2}\right)}{\left(1-P_t(\omega_t)\right)\cdot A_{d-1}(1)} \\ &= \frac{\Gamma\left(\frac{d}{2}\right)\cdot\left(1-(\cos\alpha_t\cdot\omega_t/\delta)^2\right)^{\frac{d-1}{2}}}{2\sqrt{\pi}\cdot\Gamma\left(\frac{d+1}{2}\right)\cdot\left(1-P_t(\omega_t)\right)}\end{aligned} \tag{8}$$

Where $V_d(r)$ and $A_{d-1}(r)$ are the volume of a d-dimensional ball and the area of a $d-1$-dimensional sphere with radius r, respectively. Note that the last equality comes from $V_d(r) = \frac{\pi^{d/2}}{\Gamma\left(\frac{d}{2}+1\right)}r^d$ and $A_{d-1}(r) = \frac{2\pi^{d/2}}{\Gamma\left(\frac{d}{2}\right)}r^{d-1}$. Similarly, the following holds:

$$\mathbb{E}\left[\langle \mathbf{g}_t, \mathbf{u}\rangle | \langle \mathbf{g}_t, \mathbf{u}\rangle \ge -\cos\alpha_t\cdot\omega_t/\delta\right] = \frac{\Gamma\left(\frac{d}{2}\right)\cdot\left(1-(\cos\alpha_t\cdot\omega_t/\delta)^2\right)^{\frac{d-1}{2}}}{2\sqrt{\pi}\cdot\Gamma\left(\frac{d+1}{2}\right)\cdot P_t(\omega_t)} \tag{9}$$

By using Eq. (8) and Eq. (9) in Eq. (7), we have

$$\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right] = \frac{\Gamma\left(\frac{d}{2}\right)\cdot\left(1-(\cos\alpha_t\cdot\omega_t/\delta)^2\right)^{\frac{d-1}{2}}}{2\sqrt{\pi}\cdot\Gamma\left(\frac{d+1}{2}\right)} \tag{10}$$

■

Lemma 5. *Let $\varepsilon_1, \varepsilon_2 > 0$ be given. Then the following upper and lower bounds hold for $\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t)$:*

1. *(Upper bound)*

$$\begin{aligned}\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) \le\ & \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}} \\ & + \left(1 + \frac{\sqrt{n_t}\,\beta_t}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}\right)K_{n_t,d}(\varepsilon_1, \varepsilon_2)\,.\end{aligned}$$

2. *(Lower bound)*

$$\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) \geq \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t \langle \mathbf{g}_t, \mathbf{u} \rangle\right]}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] + \varepsilon_2 + (n_t - 1)\beta_t^2 \varepsilon_1}} - \left(1 + \frac{\sqrt{n_t}\,\beta_t}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] + \varepsilon_2 + (n_t - 1)\beta_t^2 \varepsilon_1}}\right) K_{n_t,d}\left(\varepsilon_1, \varepsilon_2\right).$$

Here, the error term $K_{n_t,d}(\varepsilon_1, \varepsilon_2)$ is defined as

$$K_{n_t,d}\left(\varepsilon_1, \varepsilon_2\right) = n_t(n_t + 1) \exp\left(-Cd\varepsilon_1^2\right) + 2\exp\left(-\frac{2n_t \varepsilon_2^2}{(2\beta_t - 1)^2}\right),$$

for some universal constant $C > 0$.

Proof of Lem. 5: Let $\varepsilon_1, \varepsilon_2 > 0$ be arbitrary. We define the following sets:

$$\begin{aligned}
\mathcal{S}_i(\varepsilon_1) &\coloneqq \left\{\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_{n_t}) \mid |\langle \mathbf{g}_t, \mathbf{u}_i \rangle| \geq \varepsilon_1\right\}, \\
\mathcal{S}_{i,j}(\varepsilon_1) &\coloneqq \left\{\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_{n_t}) \mid |\langle \mathbf{u}_i, \mathbf{u}_j \rangle| \geq \varepsilon_1\right\}, \\
\mathcal{S}_\phi(\varepsilon_2) &\coloneqq \left\{\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_{n_t}) \;\middle|\; \left|\mathbb{E}[\widehat{\phi}_t^2] - \frac{1}{n_t}\sum_{i=1}^{n_t} \widehat{\phi}_{t,i}^2\right| \geq \varepsilon_2\right\}, \\
\mathcal{S}(\varepsilon_1, \varepsilon_2) &\coloneqq \left(\bigcup_{i=1}^{n_t} \mathcal{S}_i(\varepsilon_1)\right) \cup \left(\bigcup_{1 \leq i < j \leq n_t} \mathcal{S}_{i,j}(\varepsilon_1)\right) \cup \mathcal{S}_\phi(\varepsilon_2).
\end{aligned}$$

For notational convenience, we also define

$$A(\mathbf{U}) \coloneqq \sum_{i=1}^{n_t} \widehat{\phi}_{t,i} \langle \mathbf{g}_t, \delta \mathbf{u}_i \rangle.$$

Applying Lem. 2 and Lem. 3, and using the union bound, we obtain

$$\mathbb{P}\left[\mathbf{U} \in \mathcal{S}\right] \leq n_t \cdot \mathbb{P}\left[\mathbf{U} \in \mathcal{S}_1\right] + \binom{n_t}{2} \cdot \mathbb{P}\left[\mathbf{U} \in \mathcal{S}_{1,2}\right] + \mathbb{P}\left[\mathbf{U} \in \mathcal{S}_\phi\right] \leq K_{n_t,d}\left(\varepsilon_1, \varepsilon_2\right). \tag{11}$$

Now, we derive the upper bound. By the law of total probability, we have

$$\begin{aligned}
\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) &= \mathbb{E}\left[\frac{\langle \mathbf{g}_t, \widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t) \rangle}{\left\|\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right\|_2}\right] \\
&= \mathbb{E}\left[\frac{\langle \mathbf{g}_t, \widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t) \rangle}{\left\|\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right\|_2} \;\middle|\; \mathbf{U} \notin \mathcal{S}\right] \mathbb{P}[\mathbf{U} \notin \mathcal{S}] \\
&\quad + \mathbb{E}\left[\frac{\langle \mathbf{g}_t, \widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t) \rangle}{\left\|\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right\|_2} \;\middle|\; \mathbf{U} \in \mathcal{S}\right] \mathbb{P}[\mathbf{U} \in \mathcal{S}].
\end{aligned}$$

On the good event $\{\mathbf{U} \notin \mathcal{S}\}$, we expand the squared norm directly. By definition,

$$\begin{aligned}
\left\|\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right\|_2^2 &= \left\|\frac{1}{n_t}\sum_{i=1}^{n_t} \delta \widehat{\phi}_{t,i}\mathbf{u}_i\right\|_2^2 \\
&= \frac{1}{n_t^2}\left\langle \sum_{i=1}^{n_t} \delta\widehat{\phi}_{t,i}\mathbf{u}_i, \sum_{j=1}^{n_t}\delta\widehat{\phi}_{t,j}\mathbf{u}_j \right\rangle \\
&= \frac{\delta^2}{n_t^2}\sum_{i=1}^{n_t}\sum_{j=1}^{n_t}\widehat{\phi}_{t,i}\widehat{\phi}_{t,j}\langle \mathbf{u}_i, \mathbf{u}_j\rangle \\
&= \frac{\delta^2}{n_t^2}\left(\sum_{i=1}^{n_t}\widehat{\phi}_{t,i}^2 + \sum_{\substack{1\le i,j\le n_t \\ i\neq j}} \widehat{\phi}_{t,i}\widehat{\phi}_{t,j}\langle \mathbf{u}_i, \mathbf{u}_j\rangle\right).
\end{aligned}$$

From the previous expansion, we have

$$\left\|\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right\|_2 \ge \frac{\delta}{\sqrt{n_t}}\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}.$$

Thus, on the good event $\{\mathbf{U} \notin \mathcal{S}\}$, we can bound

$$\frac{\langle \mathbf{g}_t, \widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\rangle}{\left\|\widehat{\nabla S}(\mathbf{x}_t, \omega_t, \beta_t)\right\|_2} = \frac{1}{\left\|\widehat{\nabla S}\right\|_2}\left\langle \mathbf{g}_t, \frac{1}{n_t}\sum_{i=1}^{n_t}\delta\widehat{\phi}_{t,i}\mathbf{u}_i\right\rangle = \frac{1}{n_t\left\|\widehat{\nabla S}\right\|_2}A(\mathbf{U}).$$

Therefore,

$$\frac{\langle \mathbf{g}_t, \widehat{\nabla S}\rangle}{\left\|\widehat{\nabla S}\right\|_2} \le \frac{A(\mathbf{U})}{\delta\sqrt{n_t}\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}.$$

Taking expectations over the good event, we obtain

$$\mathbb{E}\left[\frac{\langle \mathbf{g}_t, \widehat{\nabla S}\rangle}{\left\|\widehat{\nabla S}\right\|_2}\,\middle|\, \mathbf{U}\notin\mathcal{S}\right] \le \frac{1}{\delta\sqrt{n_t}\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}\mathbb{E}[A(\mathbf{U})].$$

Moreover, by independence and identical distribution of the samples, we have

$$\mathbb{E}[A(\mathbf{U})] = n_t\delta\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right].$$

Therefore,

$$\mathbb{E}\left[\frac{\langle \mathbf{g}_t, \widehat{\nabla S}\rangle}{\left\|\widehat{\nabla S}\right\|_2}\,\middle|\, \mathbf{U}\notin\mathcal{S}\right] \le \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}.$$

On the bad event $\{\mathbf{U} \in \mathcal{S}\}$, we use the trivial bound

$$\left|\frac{\langle \mathbf{g}_t, \widehat{\nabla S}\rangle}{\left\|\widehat{\nabla S}\right\|_2}\right| \le 1,$$

and hence

$$\mathbb{E}\left[\frac{\langle \mathbf{g}_t, \widehat{\nabla S}\rangle}{\left\|\widehat{\nabla S}\right\|_2}\,\middle|\, \mathbf{U}\in\mathcal{S}\right] \le 1.$$

Substituting back into the law of total probability, we have

$$\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) \leq \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}(1 - \mathbb{P}[\mathbf{U}\in\mathcal{S}]) + \mathbb{P}[\mathbf{U}\in\mathcal{S}] + \frac{\sqrt{n_t}\beta_t\mathbb{P}[\mathbf{U}\in\mathcal{S}]}{\sqrt{\mathbb{E}[\widehat{\phi}_t^2] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}.$$

Grouping terms, we obtain

$$\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) \leq \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}} + \left(1 + \frac{\sqrt{n_t}\,\beta_t}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}\right)\mathbb{P}\left[\mathbf{U}\in\mathcal{S}\right].$$

Applying the bound $\mathbb{P}\left[\mathbf{U}\in\mathcal{S}\right] \leq K_{n_t,d}(\varepsilon_1,\varepsilon_2)$ from Eq. (11), we finally get

$$\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) \leq \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}} + \left(1 + \frac{\sqrt{n_t}\,\beta_t}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] - \varepsilon_2 - (n_t-1)\beta_t^2\varepsilon_1}}\right)K_{n_t,d}(\varepsilon_1,\varepsilon_2).$$

Similarly, for the lower bound, we have

$$\mu(\mathbf{x}_t, \omega_t, \beta_t, n_t) \geq \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] + \varepsilon_2 + (n_t-1)\beta_t^2\varepsilon_1}} - \left(1 + \frac{\sqrt{n_t}\,\beta_t}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] + \varepsilon_2 + (n_t-1)\beta_t^2\varepsilon_1}}\right)K_{n_t,d}(\varepsilon_1,\varepsilon_2).$$

This completes the proof of the lemma. We note that the argument does not rely on the linearity assumption.

■

Lemma 6. *Assume local linearity holds around $\mathbf{x}_t$. Then, for any constant $q \in \left[\frac{1}{2}, 1\right)$, we have*

$$\mathbb{E}\left[\widehat{\phi}\left(\mathbf{x}_t + P_t^{-1}(q)\frac{\mathbf{x}_t - \mathbf{x}^\star}{\|\mathbf{x}_t - \mathbf{x}^\star\|_2} + \delta\mathbf{u}\right)\langle \mathbf{g}_t, \mathbf{u}\rangle\right] = \Theta\left(\frac{1}{\sqrt{d}}\right).$$

Proof of Lem. 6: Based on Lem. 4, we analyze the asymptotic behavior of $\frac{\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(\frac{d+1}{2}\right)}$ and $\left(1 - (\cos\alpha_t \cdot \omega_t/\delta)^2\right)^{\frac{d-1}{2}}$ as d tends to infinity.

By Lem. 2, we have

$$\mathbb{P}\left[|\langle \mathbf{u}, \mathbf{g}_t\rangle| > \varepsilon\right] \leq 2\exp\left(-Cd\varepsilon^2\right),$$

so in particular

$$\mathbb{P}\left[\langle \mathbf{u}, \mathbf{g}_t \rangle < -\varepsilon\right] \leq \exp\left(-Cd\varepsilon^2\right).$$

We know that $1-q = \mathbb{P}\left[\langle \mathbf{u}, \mathbf{g}_t \rangle > -\cos\alpha_t \cdot \omega_t/\delta\right]$, based on the selection of the overshooting value ω_t. Thus,

$$1-q \leq \exp\left(-Cd\left(\cos\alpha_t \cdot \omega_t/\delta\right)^2\right),$$

which implies

$$\ln(1-q) \leq -Cd\left(\cos\alpha_t \cdot \omega_t/\delta\right)^2,$$

and consequently

$$1 + \frac{\ln(1-q)}{Cd} \leq 1 - \left(\cos\alpha_t \cdot \omega_t/\delta\right)^2.$$

Raising both sides to the $(d-1)/2$ power yields

$$\left(1 + \frac{\ln(1-q)}{Cd}\right)^{\frac{d-1}{2}} \leq \left(1 - \left(\cos\alpha_t \cdot \omega_t/\delta\right)^2\right)^{\frac{d-1}{2}}.$$

Applying the classical limit $\lim_{n\to\infty}\left(1+\frac{c}{n}\right)^n = e^c$, we obtain

$$\lim_{d\to\infty}\left(1 + \frac{\ln(1-q)}{Cd}\right)^{\frac{d-1}{2}} = \exp\left(\frac{\ln(1-q)}{2C}\right),$$

which implies that $\left(1 - \left(\cos\alpha_t \cdot \omega_t/\delta\right)^2\right)^{\frac{d-1}{2}} = \Theta(1)$.

On the other hand, by Stirling's approximation

$$\Gamma(n) = \sqrt{\frac{2\pi}{n}}\left(\frac{n}{e}\right)^n\left(1 + \mathcal{O}\left(\frac{1}{n}\right)\right),$$

we find that

$$\frac{\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(\frac{d+1}{2}\right)} = \Theta\left(\frac{1}{\sqrt{d}}\right).$$

Substituting these results into Eq. (10) concludes the proof.

■

Now, we proceed to prove the theorem.

Proof of Thm. 2: For any $0 < z < \frac{1}{8}$, let $n_t = d^{3z}$, $\varepsilon_1 = d^{-4z}$, and $\varepsilon_2 = d^{-z}$. Also, let $d \geq 4\frac{\mathbb{E}[\widehat{\phi}_t^2]}{\beta^2}$. We define

$$E_1 \coloneqq \frac{\sqrt{n_t}\,\mathbb{E}\left[\widehat{\phi}_t\langle \mathbf{g}_t, \mathbf{u}\rangle\right]}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] - \varepsilon_2 - (n_t-1)\beta^2\varepsilon_1}},$$

$$E_2 \coloneqq \frac{\sqrt{n_t}\,\beta}{\sqrt{\mathbb{E}\left[\widehat{\phi}_t^2\right] - \varepsilon_2 - (n_t-1)\beta^2\varepsilon_1}},$$

$$E_3 \coloneqq K_{n_t,d}(\varepsilon_1, \varepsilon_2).$$

Then using the upper bound derived in Lem. 5, we have

$$\frac{\mu(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} - 1 \leq \frac{E_1}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} - 1 + \frac{1+E_2}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} \cdot E_3. \tag{12}$$

Since $\mathbb{E}\left[\widehat{\phi}_t^2\right] = \beta^2(1-q) + (1-\beta)^2 q$ is constant, we can estimate

$$\begin{aligned}
\frac{E_1}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} - 1 &\leq \frac{1}{\sqrt{1 - \frac{\varepsilon_2 + (n_t - 1)\beta^2 \varepsilon_1}{\mathbb{E}[\widehat{\phi}_t^2]}}} - 1 \\
&\leq \mathcal{O}\left(\varepsilon_2 + (n_t - 1)\beta^2 \varepsilon_1\right) \quad \text{(since for } 0 \leq x \leq \frac{1}{2},\ \sqrt{1-x} \geq 1 - \frac{x}{2}) \\
&= \mathcal{O}(d^{-z}). \qquad (13)
\end{aligned}$$

Moreover, we have $E_1 = \Theta\left(n_t^{1/2}\right)$. Using Lem. 6, we also have $J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t) = \Theta\left(n_t^{1/2} d^{1/2}\right)$. Thus,

$$\frac{1 + E_2}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} = \Theta(d^{1/2}). \qquad (14)$$

Substituting the values of n_t, ε_1, and ε_2 into E_3 yields

$$E_3 = \Theta\left(d^{6z}\right) \exp\left(-C d^{1-8z}\right) + \exp\left(-\frac{2d^z}{(2\beta - 1)^2}\right). \qquad (15)$$

Since $1 - 8z > 0$ and exponential functions dominate polynomial growth, combining Eq. (13), Eq. (14), and Eq. (15) with Eq. (12) yields

$$\frac{\mu(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} - 1 \leq \mathcal{O}(d^{-z}).$$

Applying similar steps using the lower bound in Lem. 5, we find

$$1 - \frac{\mu(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)}{J(\mathbf{x}_t, p_t^{-1}(q), \beta, n_t)} \leq \mathcal{O}(d^{-z}).$$

Thus, the proof is complete.

■

Proof of Thm. 3 Since $J(\mathbf{x}_t, \omega_t, \beta_t, n_t)$ is increasing with respect to n_t, the optimal choice is to take n_t at its maximum allowed value:

$$n_t = \frac{c_t}{c^\star - (c^\star - 1) P_t(\omega_t^\star)}.$$

Substituting this into the definition of J and applying Lem. 4, we obtain

$$J(\mathbf{x}_t, \omega_t, \beta_t, n_t) \propto \frac{(1 - (\cos \alpha_t \cdot \omega_t / \delta)^2)^{(d-1)/2}}{\sqrt{(c^\star - (c^\star - 1) P_t(\omega_t))\, \mathbb{E}[\widehat{\phi}_t^2]}}.$$

Expanding $\mathbb{E}[\widehat{\phi}_t^2]$ gives

$$\mathbb{E}[\widehat{\phi}_t^2] = \beta_t^2 (1 - P_t(\omega_t)) + (1 - \beta_t)^2 P_t(\omega_t).$$

Thus,

$$J(\mathbf{x}_t, \omega_t, \beta_t, n_t) \propto \frac{(1 - (\cos \alpha_t \omega_t / \delta)^2)^{(d-1)/2}}{\sqrt{(c^\star - (c^\star - 1) P_t(\omega_t))(\beta_t^2 (1 - P_t(\omega_t)) + (1 - \beta_t)^2 P_t(\omega_t))}}.$$

To maximize J, it suffices to minimize

$$\beta_t^2 (1 - P_t(\omega_t)) + (1 - \beta_t)^2 P_t(\omega_t).$$

Differentiating with respect to β_t and setting the derivative to zero yields

$$\beta_t = P_t(\omega_t).$$

Substituting this optimal β_t back, we find

$$\mathbb{E}[\widehat{\phi}_t^2] = P_t(\omega_t)(1 - P_t(\omega_t)),$$

and thus the final expression to maximize is

$$\widehat{J}_t(\omega_t) = \frac{(1 - (\cos \alpha_t \omega_t / \delta)^2)^{d-1}}{P_t(\omega_t)(1 - P_t(\omega_t))(c^\star - (c^\star - 1)P_t(\omega_t))}.$$

■

Proof of Thm. 4 First, we show that $\cos \alpha_1 \geq 0$. Suppose for contradiction that $\cos \alpha_1 < 0$, i.e.,

$$\langle \mathbf{x}_1 - \mathbf{x}^\star, \nabla S(\mathbf{x}_1) \rangle < 0.$$

Using the definition of the directional derivative, we have

$$\lim_{h \to 0} \frac{S\left(\mathbf{x}_1 + h(\mathbf{x}_1 - \mathbf{x}^\star)\right) - S(\mathbf{x}_1)}{h} = \langle \mathbf{x}_1 - \mathbf{x}^\star, \nabla S(\mathbf{x}_1) \rangle < 0.$$

Since the directional derivative is strictly negative, there exists $\epsilon > 0$ such that for all sufficiently small $\epsilon > 0$,

$$S\left(\mathbf{x}_1 - \epsilon(\mathbf{x}_1 - \mathbf{x}^\star)\right) > S(\mathbf{x}_1).$$

Noting that

$$\mathbf{x}_1 - \epsilon(\mathbf{x}_1 - \mathbf{x}^\star) = \mathbf{x}^\star + (1 - \epsilon)(\mathbf{x}_1 - \mathbf{x}^\star),$$

we can rewrite this inequality as

$$S\left(\mathbf{x}^\star + (1 - \epsilon)(\mathbf{x}_1 - \mathbf{x}^\star)\right) > S(\mathbf{x}_1).$$

Since $\phi(\mathbf{x}_1) = 1$ by assumption, and assuming ϕ remains 1 in a neighborhood where S does not decrease, we also have

$$\phi\left(\mathbf{x}^\star + (1 - \epsilon)(\mathbf{x}_1 - \mathbf{x}^\star)\right) = 1.$$

Thus, for $r = 1 - \epsilon$, we find a point $0 < r < 1$ such that $\phi(\mathbf{x}^\star + r(\mathbf{x}_1 - \mathbf{x}^\star)) = 1$, contradicting the assumption that no such r exists. Therefore, our assumption that $\cos \alpha_1 < 0$ must be false, and we conclude that

$$\cos \alpha_1 \geq 0.$$

Now that we have established $\cos \alpha_1 \geq 0$, it follows that

$$\mathbb{E}[\cos \alpha_1] = \mathbb{E}\left[\cos \alpha_1 \mid \cos \alpha_1 \geq 0\right].$$

Expanding $\cos \alpha_1$ in terms of the vectors involved, we write

$$\cos \alpha_1 = \left\langle \frac{\mathbf{x}_1 - \mathbf{x}^\star}{\|\mathbf{x}_1 - \mathbf{x}^\star\|_2}, \mathbf{g}_1 \right\rangle.$$

Thus,

$$\mathbb{E}[\cos \alpha_1] = \mathbb{E}\left[\left\langle \frac{\mathbf{x}_1 - \mathbf{x}^\star}{\|\mathbf{x}_1 - \mathbf{x}^\star\|_2}, \mathbf{g}_1 \right\rangle \,\middle|\, \left\langle \frac{\mathbf{x}_1 - \mathbf{x}^\star}{\|\mathbf{x}_1 - \mathbf{x}^\star\|_2}, \mathbf{g}_1 \right\rangle \geq 0\right].$$

Finally, applying the result from Eq. (9) with $\omega_t = 0$, we obtain

$$\mathbb{E}\left[\left\langle \frac{\mathbf{x}_1 - \mathbf{x}^\star}{\|\mathbf{x}_1 - \mathbf{x}^\star\|_2}, \mathbf{g}_1 \right\rangle \,\middle|\, \left\langle \frac{\mathbf{x}_1 - \mathbf{x}^\star}{\|\mathbf{x}_1 - \mathbf{x}^\star\|_2}, \mathbf{g}_1 \right\rangle \geq 0\right] = \frac{\Gamma\left(\frac{d}{2}\right)}{2\sqrt{\pi}\,\Gamma\left(\frac{d+1}{2}\right)}.$$

This completes the proof.

■

C Empirical study of AS and AGREST

Here, we design two experiments to validate the effectiveness of our analysis for both AS and AGREST. (Fig. 4)

First, we sample 100 random correctly classified images from the ImageNet dataset and run the experiments using binary search and asymmetric search when $c^\star = 10^3$. We observe that the average cumulative search cost across iterations for binary search is approximately 2.5 times higher than that of AS. This highlights the effectiveness of AS compared to vanilla search.

Second, to show that using the overshooting value obtained by AGREST leads to a probability of making low-cost queries close to the theoretical value in Thm. 3, we again sample 100 random images and run one iteration of AGREST using 500 queries for gradient estimation. We then compute the empirical probability of making low-cost queries, defined as the ratio of low-cost to total (500) queries, and compare it to the optimal probability predicted by our theoretical analysis. As shown in Fig. 4, our analysis is close to the empirical results, especially for larger values of $c^\star$.

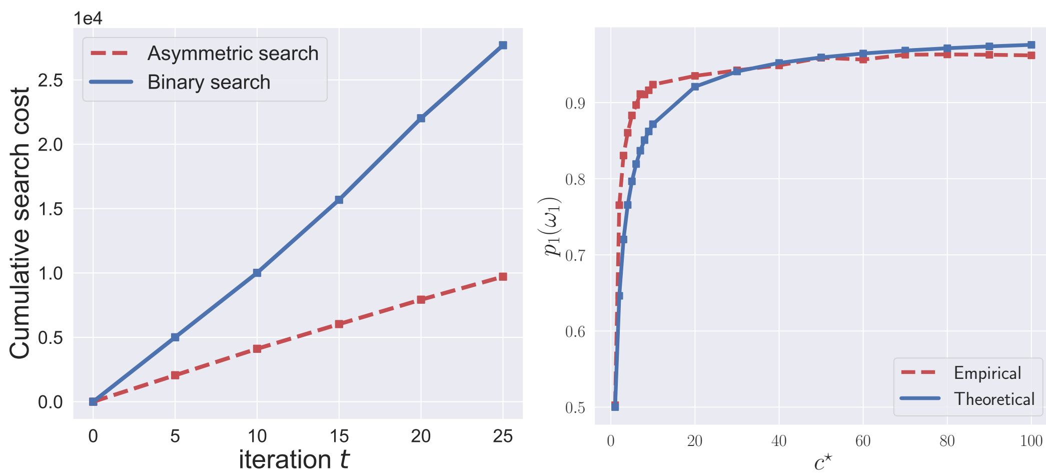

Figure 4: Empirical study of AS and AGREST. The left plot compares AS with vanilla search (binary search) in terms of cumulative *search* cost over iterations in GeoDA when $c^\star = 10^3$, while the right plot shows the optimal theoretical probability of making low-cost queries (assuming local linearity of the decision boundary) versus the empirical ratio of low-cost to total queries for different values of $c^\star$.

D Implementation

Modification to GeoDA We modify the direction-based adversarial example search phase in GeoDA. In its original implementation, GeoDA estimates the gradient and then proceeds from the original image, taking fixed-size steps along that direction until it finds an adversarial example. However, this process often leads to a large number of flagged queries, since many of the intermediate steps can cross the decision boundary.

To address this issue, we change the starting point of the search. Instead of beginning at the original image, we start from

$$\mathbf{x}'' = \mathbf{x}^\star + \|\mathbf{x}^\star - \mathbf{x}_t\|_2 \cdot \frac{\widehat{\nabla} S(\mathbf{x}_t, \omega_t)}{\|\widehat{\nabla} S(\mathbf{x}_t, \omega_t)\|_2}.$$

This new starting point lies further in the direction of the estimated gradient and is designed with the expectation that it is already adversarial—or at least closer to an adversarial example than the original image. If $\mathbf{x}''$ is not adversarial, the algorithm continues the search in the estimated gradient direction. This modification significantly reduces the number of flagged queries encountered during the search.

Selection of the hyperparameter m We select values for the hyperparameter m by evaluating the performance of the corresponding attacks under different settings of m, using 20 randomly selected correctly classified images. This evaluation is performed with $c^\star = 10^3$ and a total query cost of 250K, as shown in Fig. 5.

Although the optimal value of m can vary with $c^\star$, we choose to fix m independently of $c^\star$. This decision simplifies the attack process and avoids the additional computational overhead of tuning m for each value of $c^\star$, while still enabling effective attack performance.

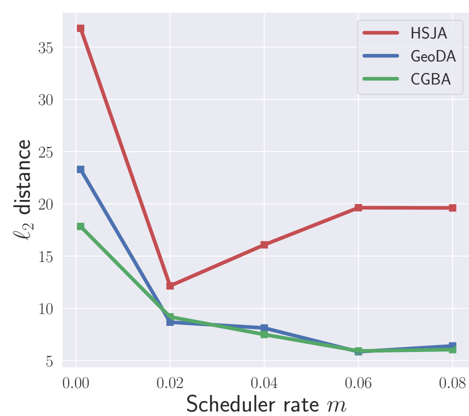

Figure 5: Median ℓ_2 distance of adversarial perturbations for varying values of m, with $c^\star = 10^3$ and a total query cost of 150K.

AGREST with dimension reduction As mentioned earlier, most practical attacks use a dimension reduction matrix $\mathbf{R} \in \mathbb{R}^{d \times d'}$ to perform the sampling process in a subspace of dimension $d' \ll d$, where d is the dimension of the original space, in order to increase sample efficiency. To apply the same subspace in the AGREST estimator, the only modifications needed compared to the original AGREST Alg. 3 are: first, projecting each sample into the subspace; and second, using the effective dimension d' to compute α_t. An overview of this version of AGREST is provided in Alg. 4.

Computation resources For our experiments on ResNet-50, we use NVIDIA P100 GPUs. All other experiments, including those involving ViT and CLIP models, are conducted on NVIDIA A100 GPUs to accommodate the higher computational and memory demands of these models.

Algorithm 4 AGREST Estimation

Inputs: Iteration t, source image $\mathbf{x}^\star$, boundary point $\mathbf{x}_t$, dimension d, sampling subspace dimension d', sampling subspace matrix $\mathbf{R}$, high-cost query cost $c^\star$, sampling radius δ, sampling batch size b, cosine value α_t, vanilla gradient estimation query budget n'_t, scheduler rate m

Outputs: Normalized approximated direction g_t, next cosine value α_{t+1}

$n_L \leftarrow 0,\ \ n_H \leftarrow 0,\ \ \mathbf{v}^+ \leftarrow \vec{0},\ \ \mathbf{v}^- \leftarrow \vec{0},\ \ \hat{c} \leftarrow 0,\ \ \omega^\star \leftarrow \textsc{Overshooting}\,(c^\star)$ ▷ Thm. 3

$\omega_t \leftarrow \omega^\star / \cos\alpha_t,\ \ c_t \leftarrow n'_t(c^\star + 1)/2$

while $\hat{c} < c_t$ **do**

  $B \leftarrow \left\{\mathbf{R}\mathbf{u}_i / \|\mathbf{R}\mathbf{u}_i\| \text{ where } \mathbf{u}_i \sim \textsc{Uniform}\left(\mathbb{S}^{d-1}\right)\right\}_{i=1}^{b}$ ▷ Dimension reduction

  for each $\mathbf{u}_i \in B$ **do**

    if $\phi\left(\mathbf{x}_t + \omega_t \frac{\mathbf{x}_t - \mathbf{x}^\star}{\|\mathbf{x}_t - \mathbf{x}^\star\|_2} + \delta \mathbf{u}_i\right) = 1$ **then**

      $\mathbf{v}^+ \leftarrow \mathbf{v}^+ + \mathbf{u}_i,\ \ n_L \leftarrow n_L + 1,\ \ \hat{c} \leftarrow \hat{c} + 1$

    else

      $\mathbf{v}^- \leftarrow \mathbf{v}^- - \mathbf{u}_i,\ \ n_H \leftarrow n_H + 1,\ \ \hat{c} \leftarrow \hat{c} + c^\star$

    end if

  end for

end while

$\hat{p}_t \leftarrow n_L / (n_L + n_H)$

$g_t \leftarrow (1 - \hat{p}_t)\,\mathbf{v}^+ + \hat{p}_t \mathbf{v}^-,\ \ \alpha_{t+1} \leftarrow \textsc{Scheduler-Step}\,(t, \hat{p}_t, m)$ ▷ Alg. 2

return $g_t / \|g_t\|_2\,,\ \ \alpha_{t+1}$

E Additional results for Vision Transformers (ViT)

In this section, we present comprehensive empirical evaluations that extend our analysis across varying budget constraints and different query cost parameters $c^\star$. Specifically, we conduct experiments utilizing Vision Transformer architectures (ViT-B/32 and ViT-B/16) on the ImageNet dataset.

Table 2: Median ℓ_2 distance for various $c^\star$ values and different types of attacks for **ViT-B/32** model on ImageNet dataset.

Attack	Method	$c^\star = 2.0$					$c^\star = 5.0$					$c^\star = 100.0$					$c^\star = 1000.0$					Higher Queries	
		Total Cost					Total Cost					Total Cost					Total Cost					Total Cost	
		1000	2000	5000	10000	15000	1000	2000	5000	10000	15000	1000	2000	5000	10000	15000	1000	2000	5000	10000	15000	150000	250000
SURFREE	VA	9.3	6.2	3.9	2.9	2.5	14.3	9.9	5.7	4.4	3.4	70.4	71.0	34.7	21.3	18.2	**69.7**	71.1	**70.6**	68.1	**68.3**	5.13	16.12
	VA+AS (A-SurFree)	**7.8**	**5.0**	**3.6**	**2.4**	**2.1**	**11.4**	**7.3**	**4.6**	**3.1**	**2.5**	**70.0**	**70.1**	**22.8**	**14.5**	**9.7**	70.1	**69.9**	71.5	**67.3**	68.5	3.68	6.35
HSJA	VA	53.7	44.8	28.7	18.3	13.9	61.6	53.2	40.0	28.1	22.8	**68.4**	**67.2**	67.5	67.9	63.1	69.9	71.1	69.3	69.4	70.2	4.21	22.46
	VA+AS	53.3	41.8	25.2	18.3	13.5	62.6	50.5	38.6	24.4	18.5	69.5	68.4	69.7	58.8	56.8	68.7	**69.4**	70.2	68.1	**68.2**	3.88	18.79
	VA+AGREST	18.6	9.7	**4.2**	2.7	2.2	36.3	**17.3**	7.8	4.4	**3.1**	69.1	71.1	73.7	**37.0**	35.4	**68.5**	72.1	**69.1**	70.1	69.7	2.19	11.28
	VA+AS+AGREST (A-HSJA)	**17.8**	**9.6**	4.7	**2.7**	**2.1**	**34.8**	19.1	**7.6**	**4.2**	3.1	71.4	69.1	**60.1**	38.3	**25.3**	71.8	72.1	71.9	**67.6**	68.7	2.06	10.74
GEODA	VA	14.2	6.6	3.4	2.7	2.4	18.9	12.0	6.7	3.5	3.2	**67.1**	**68.2**	**30.1**	26.3	18.7	69.3	**68.6**	68.1	**67.5**	69.2	3.97	9.83
	VA+AS	12.7	7.8	3.7	2.7	2.3	17.1	11.7	5.5	3.4	2.9	71.0	73.7	30.5	20.5	16.0	70.5	71.5	68.7	73.1	71.4	3.12	8.78
	VA+AGREST	**7.4**	**3.7**	2.4	**1.9**	**1.7**	**14.6**	8.8	4.5	3.3	**2.7**	70.2	69.8	36.6	23.5	17.7	**68.6**	72.2	**66.0**	69.0	69.5	2.10	5.12
	VA+AS+AGREST (A-GeoDA)	7.8	3.9	**2.3**	1.9	1.8	14.8	**8.4**	**4.4**	**3.1**	2.8	68.8	70.6	34.0	**19.0**	**15.6**	72.2	72.5	73.5	72.9	**68.2**	2.03	4.35
CGBA	VA	11.6	6.0	3.0	1.6	1.4	18.4	11.3	5.3	3.1	2.1	70.4	72.0	28.0	24.5	17.0	**68.8**	**70.4**	69.7	72.8	72.8	2.13	9.67
	VA+AS	10.8	6.4	3.2	1.7	1.3	16.2	10.7	4.6	**2.6**	**2.0**	68.5	73.4	29.2	18.1	13.7	71.9	73.8	68.3	**68.5**	70.1	1.97	8.24
	VA+AGREST	**7.3**	**3.6**	2.0	1.5	1.3	15.8	8.3	**4.0**	3.0	2.5	**67.1**	73.5	39.2	18.9	15.2	70.9	70.5	72.7	69.8	**69.5**	1.56	5.46
	VA+AS+AGREST (A-CGBA)	7.6	4.1	**1.9**	**1.5**	**1.2**	**14.4**	**7.7**	4.2	2.9	2.5	70.2	**67.2**	**23.9**	**15.9**	**13.4**	71.9	70.4	**67.3**	70.0	72.7	1.42	5.61

Table 3: Median ℓ_2 distance for various $c^\star$ values and different types of attacks for **ViT-B/16** model on ImageNet dataset.

Attack	Method	$c^\star = 2.0$					$c^\star = 5.0$					$c^\star = 100.0$					$c^\star = 1000.0$				
		Total Cost					Total Cost					Total Cost					Total Cost				
		1000	2000	5000	10000	15000	1000	2000	5000	10000	15000	1000	2000	5000	10000	15000	1000	2000	5000	10000	15000
SURFREE	VA	10.7	7.1	4.2	3.0	2.3	16.6	11.0	6.8	4.4	3.6	**57.6**	**57.1**	33.3	20.7	18.6	60.0	56.0	56.8	**56.8**	**57.6**
	VA+AS (A-SurFree)	**8.9**	**6.0**	**3.7**	**2.3**	**2.0**	**14.4**	**8.3**	**4.6**	**3.2**	**2.6**	58.2	58.2	**27.7**	**15.0**	**10.4**	**58.0**	**54.9**	**54.6**	56.9	58.3
HSJA	VA	37.5	29.9	17.8	10.8	7.1	49.6	40.0	26.0	18.4	13.9	57.3	**56.0**	55.1	52.5	49.7	56.7	58.9	59.2	57.2	58.0
	VA+AS	38.1	28.6	18.5	9.6	6.8	47.8	38.5	23.9	16.5	11.0	57.4	56.6	55.0	47.4	44.5	56.6	60.9	57.9	59.2	**57.4**
	VA+AGREST	**14.5**	8.0	4.0	**2.3**	**1.7**	29.4	14.6	**6.0**	3.8	2.7	**56.7**	56.9	59.6	**30.5**	31.6	**55.9**	**55.3**	58.2	57.4	59.0
	VA+AS+AGREST (A-HSJA)	15.1	**7.7**	**3.9**	2.3	1.8	**27.1**	**14.5**	6.4	**3.6**	**2.6**	58.1	58.1	**40.5**	31.3	**21.0**	58.2	56.7	**57.5**	**56.7**	58.7
GEODA	VA	12.5	7.3	3.1	2.2	1.8	19.7	13.7	6.5	3.3	2.5	56.8	57.7	32.5	25.5	22.6	58.2	**56.5**	60.3	**57.4**	**55.8**
	VA+AS	14.4	7.1	3.2	2.0	1.9	19.9	12.6	5.5	3.3	2.5	60.2	60.6	30.6	23.1	17.6	**58.0**	57.0	**58.0**	59.3	57.3
	VA+AGREST	8.4	4.0	**2.1**	**1.5**	1.4	**13.5**	**8.1**	**3.8**	2.6	**2.1**	58.3	**55.8**	35.5	19.4	17.0	58.8	59.0	58.4	57.6	57.4
	VA+AS+AGREST (A-GeoDA)	**7.9**	**3.8**	2.1	1.6	**1.4**	13.7	8.5	4.1	**2.5**	2.2	**55.9**	58.5	**27.5**	**18.1**	**15.2**	59.2	60.6	58.4	57.7	58.0
CGBA	VA	11.3	6.0	2.3	1.3	1.0	16.6	11.2	4.6	2.4	1.6	57.1	**54.9**	32.4	23.7	18.6	**56.1**	56.7	59.5	61.3	59.2
	VA+AS	11.2	5.6	2.4	1.3	1.0	16.4	9.5	4.3	2.2	**1.5**	56.2	56.6	28.1	17.2	14.8	59.0	58.0	59.9	58.3	55.5
	VA+AGREST	**6.5**	**3.4**	1.8	**1.2**	**0.9**	14.0	7.9	3.8	2.2	1.8	**55.0**	55.7	35.0	19.5	14.1	56.3	**55.4**	59.3	**57.6**	**53.5**
	VA+AS+AGREST (A-CGBA)	7.6	3.5	**1.7**	1.2	1.0	**12.2**	**7.5**	**3.4**	**2.2**	1.9	58.1	60.5	**21.5**	**16.5**	**12.2**	58.2	58.2	**55.9**	58.8	57.2

F Asymmetric Search (AS) illustration

Fig. 6 provides a visual example of the Asymmetric Search (AS) algorithm running with parameters $\tau = 0.1$ and $c^\star = 2$. The illustration shows the iterative progression and query evaluations leading to successful convergence near the decision boundary.

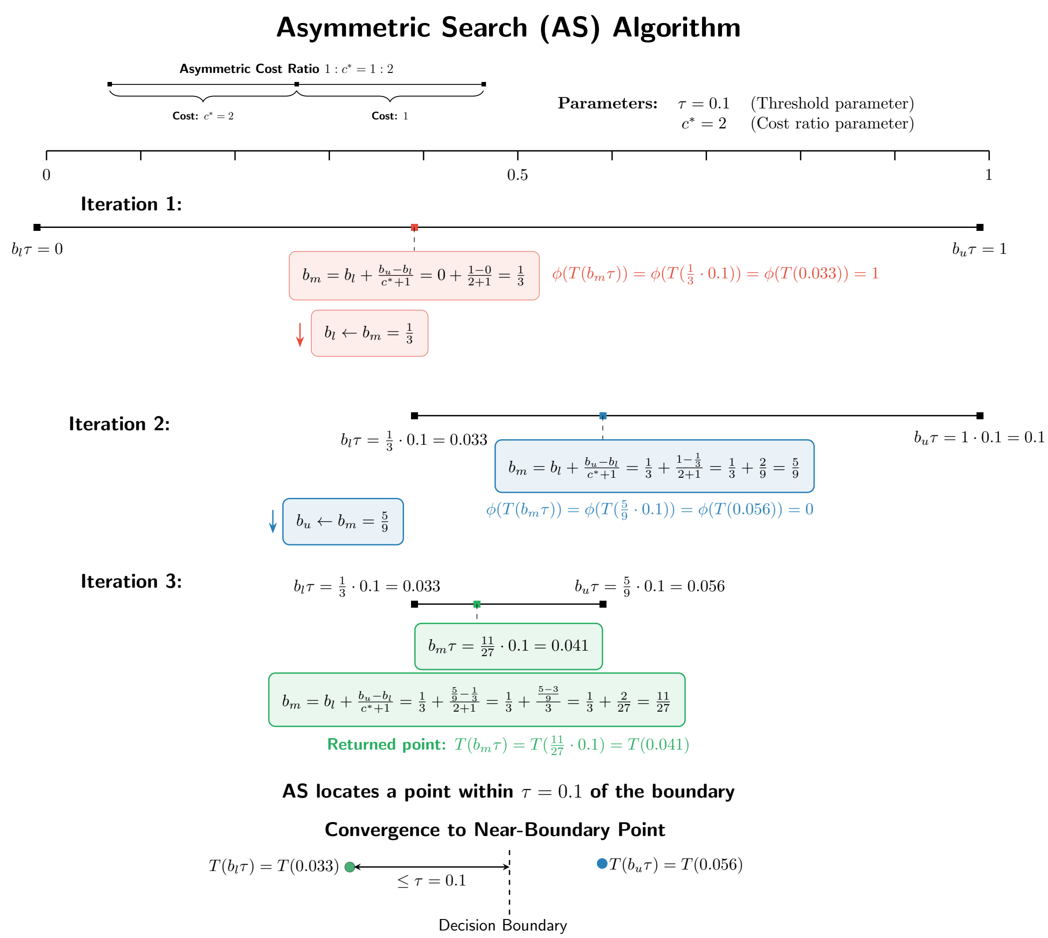

Figure 6: Asymmetric Search (AS) illustration.

G Asymmetric attacks against CLIP

We evaluate the robustness of vision-language models (VLMs), such as CLIP [28], against **stealthy** adversarial attacks. Our experiments cover both the zero-shot and fine-tuned versions of CLIP. We apply our asymmetric attacks to these models and observe substantial improvements over stealthy baselines. As shown in Fig. 7, after making 300 total queries, asymmetric methods achieve 40–60% lower ℓ_2 distortion compared to Stealthy HSJA.

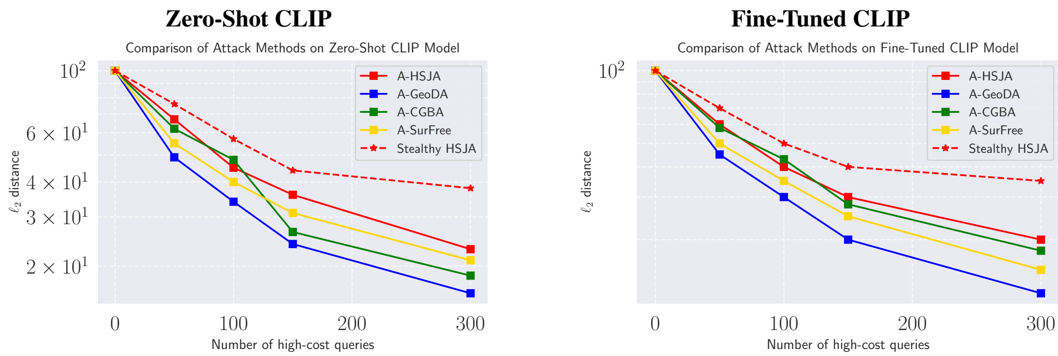

Figure 7: Performance of various asymmetric attacks compared to Stealthy HSJA on CLIP.

H Conceptual Illustration

In this section, we show the conceptual illustration of the vanilla gradient estimation and our proposed gradient estimation AGREST.

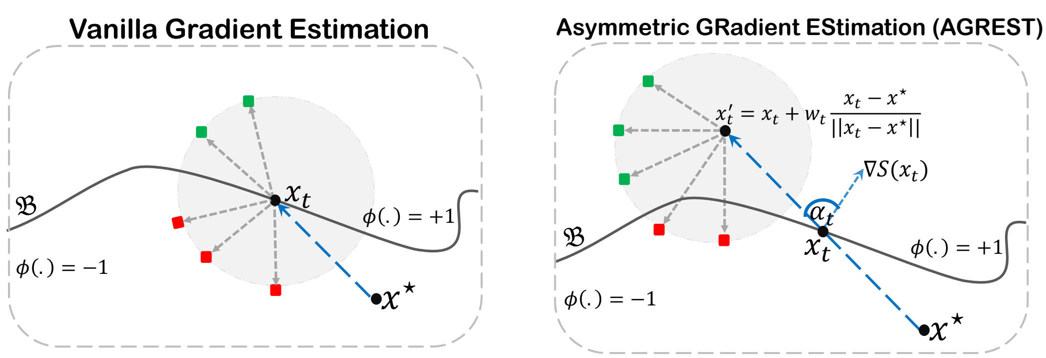

Figure 8: **Comparison of vanilla gradient estimation and its asymmetric counterpart.** Vanilla sampling results in roughly half high-cost and half low-cost queries, whereas AGREST reduces the frequency of high-cost queries by shifting the sampling region and weighting outcomes accordingly.