2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

AS-ES: Sparse Black-box Adversarial Attack By Active Subspace Evolution Strategy

Jinling Duan[1] and Zhenhua Li[2][0000-0002-6212-6384]

[1] School of Computer Science and Technology
[2] Nanjing University of Aeronautics and Astronautics, Nanjing, China
zhenhua.li@nuaa.edu.cn

Abstract. In adversarial attacks, most existing methods adopt global attack methods, which attack by changing all image pixels, but this is not realistic. On the contrary, sparse attacks indicate that only perturbing local regions of the input image can deceive DNN models into making incorrect predictions. However, this method requires a large number of queries to generate adversarial examples, and the key issues it faces are locating the perturbation area and optimizing the magnitude of the perturbation. Currently, generating high-quality adversarial examples and improving query efficiency in restricted environments is a challenge for black-box attacks. In this paper, we propose a sparse black-box attack method based on the Active Subspace Evolution Strategy (AS-ES), which locates the active subspace of the input image through the multi-arm bandit method, and uses the Covariance Matrix Adaptive Evolution Strategy algorithm for perturbation search in the low-dimensional subspace. We model this problem as a bi-level optimization problem, optimizing both the perturbation position and magnitude to generate high-quality adversarial examples while achieving efficient attacks. We conducted extensive experiments on multiple datasets and verified that the AS-ES method generates adversarial examples with higher quality and query efficiency than existing state-of-the-art attack methods.

Keywords: Adversarial Example, Black-box Attack, Sparse Perturbation, Active Subspace, Covariance Matrix Adaptive Evolution Strategy.

1 Introduction

Deep neural networks (DNNs) have demonstrated exceptional performance across various domains, including image classification [1], natural language processing [2], autonomous driving [3], and face recognition [4]. However, recent studies have shown that deep learning models are highly susceptible to adversarial attacks. Adversarial examples (AEs) can deceive DNNs into making incorrect predictions by introducing subtle, human-imperceptible perturbations to input images [5]. This phenomenon exposes the vulnerabilities of deep learning models in real-world applications.

To evaluate and improve the robustness of DNNs, researchers worked on developing more powerful adversarial attack methods, which can promote the development of more effective model defense mechanisms [6]. The key issues in this field include effectively

generating high-quality adversarial examples and improving the query efficiency and success rate of black-box attacks. The quality of adversarial examples is usually constrained by different norms, such as ℓ_0, ℓ_1, ℓ_2 and ℓ_∞-norm. A smaller ℓ_p value means that the generated perturbation is more difficult to detect, thereby increasing the stealthiness of the attack. Considering generate AEs requires a high query cost in black-box attacks, generating high-quality adversarial examples within a limited number of queries has become a key challenge. In addition, a high success rate of attacks indicates that attackers are able to make the target model misclassify adversarial examples with a high probability, this reflects the vulnerability of the model.

Researchers have made significant progress in studying black-box attacks, such as the state-of-the-art (SOTA) Square attack [7], which can significantly reduce the number of queries for black-box attacks under ℓ_∞ and ℓ_2 constraints. The perturbation generated by this method is a global perturbation, which can usually improve the success rate of attacks [8]. However, adding this perturbation to the original image will result in significant visual differences, and the quality of the adversarial examples is very low [9,10]. There are many existing studies on sparse attacks that can fool deep learning models by perturbing local regions of the original image [11]. The key to sparse attacks is to determine the location of perturbations and optimize their magnitude [12]. Most existing adversarial attacks mainly optimize the magnitude of perturbations, making them imperceptible, but the location of perturbations is all pixels of the input image, which results in low efficiency in searching for adversarial perturbations in high-dimensional space [13].

In adversarial attacks, there is a trade-off between perturbation norm and query numbers. To address this issue, a natural idea is that we can guide the black-box attack search to the active subspace of the input image, which can most significantly affect the predicted output of the target model, thereby realizing perturbation search in a low-dimensional space, greatly improving the efficiency of search and query. Therefore, we propose a score-based black-box sparse attack method. We divide the input image into multiple subspaces and sample in different subspaces. By adaptively adjusting the sensitivity and importance of the subspaces, we select the best subspace to generate high-quality adversarial examples while achieving efficient attacks. The key idea is to select a low dimensional subspace and perform search perturbation within that region, constraining the perturbation norm within a small range. The selection of subspaces and the generation of perturbations follow a dynamic learning process, where the chosen subspaces are evaluated based on the function loss values of the adversarial examples, allowing the model to learn and identify active subspaces.

We propose a sparse black-box attack method to learn and obtain the active subspace of input images and generate high-quality adversarial examples, which is called the active subspace evolution strategy (AS-ES) method. In summary, we make the following contributions in this work.

- We propose an effective black-box attack method, called Active Subspace Evolution Strategy (AS-ES), which dynamically learns the active subspace and optimizes the perturbation magnitude in this sensitive region through a bi-level optimization mechanism.

- We explored sparse perturbation search in low-dimensional subspaces and greatly improved the search efficiency by using the covariance matrix adaptation evolutionary strategy(CMA-ES). We also designed an effective active subspace evaluation method to evaluate the sampled subspace by optimizing the search information, thereby learning the active subspace.
- Extensive experiments show that the proposed AS-ES attack method achieves the expected results on multiple datasets and CNN models, and its attack success rate and query efficiency are better than the state-of-the-art black-box attack methods. We also performed ablation experiments to select the most effective experimental parameters.

2 Related Work

In this section, we give a general representation of adversarial attacks. Next, we introduce relative research on sparse attack and global attack. Finally, we explain the research motivation of this work.

2.1 Adversarial Attack

Researching adversarial examples can improve the robustness and stability of deep learning models. In image classification tasks, given a well-trained DNN classifier $H(x) = \text{argmax}_i F(x)_i$, where $x \in [0,1]^{\dim(x)}$ is an input to the neural network $F(\cdot)$, and n is the number of classes $i = 1, \dots, n$.

When the DNN correctly classifies the input image, i.e., $H(x) = y$, where y is the ground-truth label of the input x. When conducting adversarial attacks, the attacker's goal is to find an adversarial example x_{adv} that is very close to the original input x. Even if the perturbation δ is very small and imperceptible, it can trick the deep neural network into making incorrect classification predictions, i.e., $H(x_{adv}) \neq y$. We define the distance between adversarial examples and the original image as $\| x - x_{adv} \|_p \leq \epsilon$, This is usually restricted with a perturbation norm ℓ_p, $p = 0,1,2,\infty$, this is an important indicator used to measure the quality of adversarial examples. By using the perturbation magnitude as a constraint for the AEs, and then maximizing the confidence distance between the AEs and the original image x, thus generate adversarial examples that are as similar as possible to the original image. We can construct the generation of adversarial examples as the following optimization problem:

$$\max_{\delta} |H(x+\delta) - H(x)|, s.t. \| \delta \|_p \leq \epsilon \quad (1)$$

2.2 Sparse Attack and Global Attack

According to the different perturbation regions, adversarial perturbations can be divided into global and local perturbations. Global perturbation attacks usually change all pixels of the image [9]. This attack method can achieve a high attack success rate, but it requires perturbation search in high dimensions, and the scope of modification is

too large [10], resulting in obvious visual differences between the generated AEs and the original images. Existing global attacks [14] usually rely on optimizing the magnitude of the perturbation to ensure the minimum perception of the perturbation, thereby achieving an attack on the target model, but the adversarial examples have low quality.

Recently, sparse attacks have become a hot topic in adversarial attack research as a local perturbation strategy. Unlike traditional global perturbation attack methods, sparse attacks [15] indicate that by perturbing only a portion of the input image, DNNs can make incorrect classification results. However, the key challenge for sparse attacks is to determine the location of perturbations and to optimize the magnitude of perturbations at these locations [12]. Existing sparse attack methods are usually divided into three types: manual attack, heuristic attack, and optimization attack. Manual attacks are often achieved by adding visible local patches to the image [16], which lacks automation and is inefficient. Heuristic attacks, such as Jacobian-based saliency map attacks [17], use salient regions of the image to select perturbation positions, but they also suffer from poor localization accuracy and high computational complexity. Optimization attacks search for the optimal perturbation location through optimization algorithms, such as one pixel attack [18], which use differential evolution algorithms to search for one pixel and only perturb in one pixel. Current sparse attack methods focus on optimizing perturbation magnitude to ensure imperceptibility [19], but their efficiency in searching for the optimal perturbation region in high-dimensional spaces remains a major challenge.

2.3 Motivation

Although existing sparse attacks have made some significant progress, it is often difficult to balance the localization of salient regions in the input image and minimize the perturbation ℓ_p norm. Most sparse attack methods tend to prioritize query efficiency without limiting the magnitude of the perturbation [20], or achieve imperceptibility of the perturbation by constraining ℓ_p norm [21], but this requires large queries.

To address the issues of sparse attacks, a natural idea is to find a scientifically effective method to locate the active subspace of the input image, guide adversarial attacks to perturb sensitive areas, while constraining and optimizing the magnitude of perturbations, and efficiently generate adversarial examples by finding suitable sampling perturbations. Therefore, we propose an active subspace evolution strategy(AS-ES) algorithm aimed at dynamically learning the active subspace of the input image. We design a subspace evaluation mechanism to evaluate and update the sensitivity of each region, and use a covariance matrix adaptive evolution strategy(CMA-ES) to simulate the local geometric shape of the search, reducing the dimensionality of the search space and achieving higher query efficiency.

3 Proposed Method

In this section, we introduce the proposed method called AS-ES attack, which efficiently generates high-quality adversarial examples.

3.1 Multi-Arm Bandit Model for Subspace Attack

Subspace Attack Objective Function. In the adversarial attack task, we let $H(x) = \arg max_i F(x)_i$ denote the maximum logit value of the target model, where $F(x) \in \mathbb{R}^n$ represents the softmax output for input $x \in [0,1]^d$. The adversarial attack aims to find a perturbation vector $\delta \in \mathbb{R}^d$. Inspired by Carlini and Wagner's work, We set the objective loss function to minimize the distance between the logit values of the target class and the second largest class [22].

$$\min_\delta \mathcal{L}(x+\delta) = H_y(x+\delta) - \max_{j \neq y} H_j(x+\delta),\ s.t. \parallel \delta \parallel_p \leq \epsilon \tag{2}$$

where $H_j(\cdot)$ denotes the logit value for class j, y is the target class, ϵ controls the perturbation magnitude.

Global adversarial perturbations often suffer from high ℓ_p-norms that degrade the quality of adversarial examples. Therefore, we propose a sparse attack method that operates on active subspaces. Research has shown that not all input image subspaces can generate adversarial examples. In contrast, some of these subspaces are more active and more likely to generate high-quality adversarial examples. We abuse notation slightly to better illustrate this issue. δ denotes the generated global perturbation, δ_B denotes the sparse perturbation on the subspace B. Thus the above problem can be transformed into an attack problem based on the active subspace.

$$\max_{\delta_B \in B} |H(\mathrm{x}+\delta_B) - H(\mathrm{x})|, s.t. \parallel \delta_B \parallel_p \leq \epsilon \tag{3}$$

Establish Multi-Arm Bandit Model. We divide the input image $x \in \mathbb{R}^{3\times H\times W}$ into K non-overlapping subspaces $\mathcal{B} = \{B_i\}_{i=1}^K$, where each subspace $B_i \subset \mathbb{R}^{3\times m\times m}$ has dimensions satisfying $m \ll \min(H, W)$. At each iteration t, we select a subspace $B_i \in \mathcal{B}$ from K subspaces for perturbation search. The simplest method is to sequentially access one subspace at a time, but this method requires a large number of queries and has very low efficiency in generating adversarial examples. In addition, the number of K usually increases exponentially with the increase of input image dimensions, and the algorithm cannot sample every subspace. Therefore, we will model the sparse black-box attack problem based on active subspaces as a multi-arm bandit (MAB) problem. We can describe this problem as a bi-level optimization problem expressed as follows:

$$\begin{aligned} &\text{(Outer)} \quad B^* = \arg\max_{B_i \in \mathcal{B}} \mathcal{S}(B_i) \\ &\text{(Inner)} \quad \delta_{B_i}^* = \arg\max_{\delta_{B_i}} \left| H\left(x+\delta_{B_i}\right) - H(x) \right| \\ &\qquad\qquad \text{s.t.} \ \parallel \delta_{B_i} \parallel_p \leq \epsilon \end{aligned} \tag{4}$$

where $\mathcal{S}: \mathcal{B} \to \mathbb{R}^+$ quantifies the activity level of subspace B_i, updated dynamically through the average perturbation magnitude of the objective function loss within the subspace B_i.

$$B^* = \arg\max_{B_i \in \mathcal{B}} \mathbb{E}_{\delta \in B, \delta \sim \mathcal{N}(0, I_B)} [\mathcal{L}(\delta)]. \tag{5}$$

We establish a sampling probability $p_i \in \mathbb{P}$ for each subspace B_i, where $\mathbb{P} = [p_1, \dots, p_K]^T$ and $\sum_{i=1}^{K} p_i = 1$. We employ an Upper Confidence Bound (UCB) strategy to dynamically select N subspaces per iteration ($N \ll K$), balancing exploration of uncertain regions and exploitation of known high-potential areas. Then, based on optimized search information, evaluate the sampled subspaces B_i and update the sampling probability p_i, so that the probability of subspaces that meet the requirements is sampled increases, thereby we can learn to obtain the active subspaces.

3.2 Active Subspace Evolution Strategy

Generate Sparse Perturbations. In black-box attacks, only the output information of the target model can be obtained through queries, with very limited accessible information. Therefore, finding the correct search direction can effectively improve attack efficiency. We employ the Covariance Matrix Adaptation Evolution Strategy (CMA-ES) to explore random perturbations.

We propose generating sparse adversarial perturbations within an active subspace. Given an input image partitioned into K disjoint regions $\{B_i\}_{i=1}^{K}$, we maintain sparsity by generating perturbations only in $\lambda \ll K$ selected regions per iteration. The CMA-ES algorithm is mainly implemented by controlling the mean m, step size σ, and covariance matrix C. It can simulate the local geometric shape of the search space, achieving higher search efficiency. In CMA-ES, each iteration generates λ candidate solutions from $N(m_t, \sigma_t^2 C_t)$, sampled in the following form, where each $y_i \in R^n$ is a search direction.

$$\delta_i \sim m_t + \sigma_t y_i, y_i \sim N(0, C_t), i = 1, \dots, \lambda \tag{6}$$

Since the sampled mean m_t and y_i are global, the perturbation δ generated by the CMA-ES algorithm is of the same size as the input image x. To ensure that the obtained perturbation is sparse, we need to project the generated global perturbation δ_i onto the sampled subspace B_i, retaining only the perturbations within that subspace to obtain δ_{B_i}, while setting the perturbations in the unsampled coordinate region to 0 to obtain a sparse perturbation.

$$\delta_{B_i} = Proj_{B_i}(\delta) \tag{7}$$

$$\delta = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_n \end{bmatrix}, \quad \delta_{B_i} = \begin{cases} \delta_j, j \in B_i \\ 0, j \notin B_i \end{cases} \tag{8}$$

Evaluation of Subspace. At each iteration, we perform λ sampling on all regions of the input image x. Utilizing evolutionary strategy algorithm to generate sparse adversarial perturbations δ_{B_i} in each subspace B_i. Adding the sparse perturbation to the original image x to obtain an adversarial example x'. We evaluate the generated adversarial examples using the $C\&W$ loss to compute the objective function value for the new solution.

$$\mathcal{L}(x'_{1:\lambda}) \leq \mathcal{L}(x'_{2:\lambda}) \leq \cdots \leq \mathcal{L}(x'_{\lambda:\lambda}) \tag{9}$$

To determine the active subspace of perturbation, we established a subspace evaluation mechanism that rewards the sampled subspace based on the ranking of the loss function $\mathcal{L}(x')$ of adversarial examples. We define the subspace reward for sampling selection as r_i. For the rank $rank_i$ ($rank_i = 1$ is optimal), the higher the subspace ranking, the higher the reward given.

$$r_i^{(t)} = \frac{\lambda - rank_i}{\sum_{j=1}^{\lambda} j},\ i \in 1, \dots, K \tag{10}$$

We use a sliding window mechanism to store the rewards corresponding to the last t iterations in the subspace, ignoring the old reward information stored in historical iterations. Each subspace B_i maintains a reward history queue $R_i^{(t)}$ with a length not exceeding W, representing the reward history sequence of the i_{th} subspace at the t_{th} iteration, denoted as $[r_{i,1}^{(t)}, r_{i,2}^{(t)}, \dots, r_{i,n}^{(t)}]$, and $i \in 1, \dots, K$. When the number of information stored in the reward history queue exceeds the size of the sliding window W, we remove the old data from the queue head and add the new reward $r_i^{(t)}$ corresponding to the subspace from the queue tail.

$$R_i^{(t+1)} = \begin{cases} Proj_{tail(W-1)}\left(R_i^{(t)}\right) \oplus r_i^{(t)} & \text{if } |R_i^{(t)}| \geq W \\ R_i^{(t)} \oplus r_i^{(t)} & \text{otherwise} \end{cases} \tag{11}$$

where the $r_i^{(t)}$ represents the newly observed reward value for the i_{th} subspace in the current iteration, $tail(n)$ represents taking the last n elements in the sequence (removing the first element of the sequence), $\oplus$ is an append operation, that is $[a, b] \oplus c = [a, b, c]$.

In addition, we calculate the average reward for each subspace based on the reward history queue information mentioned above.

$$\bar{R}_i^{(t+1)} = \begin{cases} \mathbb{E}[R_i^{(t+1)}] & \text{if } R_i^{(t+1)} \neq 0 \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Combining historical performance with current rewards, we use Exponential Moving Averages (EMA) with momentum factor $\alpha = 0.9$ to balance new and old Information and update the sampling probability of each subspace.

$$p_i^{(t+1)} = \alpha p_i^{(t)} + (1 - \alpha) \cdot \bar{R}_i^{(t+1)} \tag{13}$$

This balances long-term performance and short-term rewards, enabling both convergence in stationary environments and dynamic tracking in non-stationary scenarios.

We introduced the epsilon-greedy mechanism to select the subspaces. We dynamically adjust the value of ϵ on based on the current search situation. At the beginning, we set $\epsilon = \epsilon_{max} = 1$, and as the number of iterations t increases ϵ gradually decays to ϵ_{min}.

$$\epsilon^{(t+1)} = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min}) \cdot e^{-\eta t} \tag{14}$$

where η is the attenuation coefficient, the larger η is, the faster the attenuation. ϵ_{min} ensure long-term exploration probability. This article set $\epsilon_{min} = 0.1, \eta = 0.01$.

Sampling of Subspace. We formulate subspace sampling as a multi-armed bandit problem solved through Upper Confidence Bound (UCB) optimization. Each subspace B_i maintains an adaptive score combining historical performance and uncertainty estimation, which can quantify the potential of each subspace (upper bound of reward expectation), and prioritize selecting the subspace with the highest potential.

$$UCB_i = \bar{R}_i + \sqrt{\frac{2\ln N}{N_i}} \tag{15}$$

where $\bar{R}_i$ denotes empirical reward average. The second term is the confidence interval term, which is used to measure the uncertainty of subspace. N_i represents the number of sampling times in the i_{th} subspace, and N represents the total number of sampling selections for all K subspaces. To generate the selection probability, we apply temperature-controlled softmax normalization.

$$P_{ucb}^{(t+1)} = \frac{\exp(UCB_i/\tau)}{\sum_{j=1}^{K} \exp(UCB_j/\tau)} \tag{16}$$

The temperature parameter τ dynamically regulates exploration and exploitation trade-off. A higher τ promotes uniform exploration while lower τ focuses on high-reward subspaces. Through iterative updates, this mechanism spontaneously converges to optimal subspace distributions.

The sampling of subspaces requires a balance between exploration and exploit mechanisms. At each iteration, with probability ϵ, we perform explorative sampling from the UCB-optimized distribution. prioritizing subspaces with high uncertainty-adjusted rewards.

$$k \leftarrow P_{\text{explore}}^{(t+1)} = P_{\text{ucb}}^{(t+1)},\ k \in 1, \ldots, K \tag{17}$$

Conversely, with probability $1 - \epsilon$, we exploit historical knowledge by sampling from the normalized EMA-updated probabilities.

$$k \leftarrow P_{\text{exploit}}^{(t+1)},\ P_{\text{exploit}}^{(t+1)} = \frac{p_i^{(t+1)}}{\sum_{i=1}^{K} p_i^{(t+1)}} \tag{18}$$

Intuitively, active subspaces are more likely to generate high-quality random perturbations. By continuously iterating, the sampling probability of this subspace will become higher, prompting the algorithm to focus on searching within such subspaces, thereby achieving the learning of active subspaces.

Update the Evolutionary Strategy Parameters. We consider the solutions with the top μ objective function values $\mathcal{L}(x)$ as the better perturbations for this iteration, and they are more likely to obtain high-quality adversarial examples. We perform a weighted summation of these μ sampled data $[\delta_i, \dots, \delta_\mu]$ to update the sampling mean of the evolutionary strategy. In the algorithm, the sum of the weights of all the sampled data is 1, and $w_1 \geq w_2 \geq \cdots \geq w_\mu > 0$ is usually taken to emphasize those candidate solutions that are ranked at the top.

$$\boldsymbol{m}^{(t+1)} = \sum_{i=1}^{\mu} w_i \, \delta_{i:\lambda}, \sum_{i=1}^{\mu} w_i = 1 \tag{19}$$

The update of the covariance matrix is crucial as it determines the shape of the sampling distribution and the search direction for perturbations. In black box attack scenarios, the dimensions of the search space typically range from $10^3 \sim 10^6$, which is far beyond the scope of covariance matrix calculations. Therefore, to deal with high dimensions, we adopt a simple rank-1 evolutionary strategy in CMA-ES. In addition, we set the appropriate step size σ and then update the evolutionary path $p_i^{(t+1)}$ and covariance matrix $C_i^{(t+1)}$.

$$\boldsymbol{p}_i^{(t+1)} = (1 - c_c)\boldsymbol{p}_i^{(t)} + \sqrt{c_c(2 - c_c)\mu_{eff}} \frac{\boldsymbol{m}^{(t+1)} - \boldsymbol{m}^{(t)}}{\sigma^{(t)}} \tag{20}$$

$$\boldsymbol{C}_i^{(t+1)} = (1 - c_1)\boldsymbol{C}_i^{(t)} + c_1 \boldsymbol{p}_i^{(t+1)} \boldsymbol{p}_i^{(t+1)^T} \tag{21}$$

3.3 Algorithm of AS-ES

In the following, we will give the details of the AS-ES algorithm.

Initialization: Firstly, we initialize the required variables. Set the CMA-ES evolutionary strategy population size to $\lambda = 4 + 3\lfloor \ln 3 * d * d \rfloor$, $\sigma = 0.5$ as the coordinate wise standard deviation, the learning rate for updating the covariance matrix is $c_{cov} = 0.01$, Initialize the subspace with sides of length $m = 3$, Since the input image size is $x \in R^{3*d*d}$, then we set the total number of subspaces divided by the original image is $K = \lfloor (d/m)^2 \rfloor$.

Select the active subspace: At each iteration t, we use the Upper Confidence Bound algorithm to select $\lambda \ll K$ from all subspaces. The initialization region selection probability $p_i \in \mathbb{P}$ is uniformly distributed, meaning that the probability of sampling each region is equal, set to $p_i = 1/K$. We introduce the epsilon-greedy mechanism for subspace sampling. To balance exploration and exploitation, with a probability of ϵ, randomly select a subspace from the P_{ucb} distribution. In addition, with a probability of $1 - \epsilon$, sample subspace from the normalized EMA-updated probabilities P utilizing known information.

Algorithm 1 AS-ES

Input: image $x \in [0,1]^{3\times d\times d}$, objective function $\mathcal{L}$, classifier $H(\cdot)$, c_c, σ,c_1, subspace side length m, windows size W;

Output: perturbation $\boldsymbol{\delta}_B$;

Initialization: $t = 0$, $K = \lfloor (d/m)^2 \rfloor$, λ, $\mathbf{m}^t = 0$, $\mathbf{C}^t = \mathbf{I}$, $\mathbf{p}^t = 0$, $p_i = 1/K$, $N = 0$, α, ϵ_{max}, ϵ_{min}, η;
while $H(x) = H(x + \delta_B)$ and $t \leq T$ **do**
  //Sampling of Subspace
  for $i \leftarrow 0$ *to* λ **do**
    if $g \sim U(0,1)$ and $g < \epsilon^{(t)}$ **then**
      sample $k \leftarrow P_{\text{ucb}}^{(t)} = \text{Softmax}(\tau \cdot \text{UCB})$
    else
      sample $k \leftarrow P^{(t+1)}$, $P^{(t+1)} = \frac{p_i^{(t+1)}}{\sum_{i=1}^{K} p_i^{(t+1)}}$.
    end if
    $N_k = N_k + 1$
  end for
  //Generate Sparse Perturbations
  for $i \leftarrow 0$ *to* λ **do**
    $\delta_i \sim m^t + \sigma^t y_i$, $y_i \sim N(0, C^t)$
    $\delta_{B_k} = Proj_{B_k}(\delta_i)$
    $x' = x + \delta_{B_k}$
    sort x' as $\mathcal{L}(x'_{1:\lambda}) \leq \mathcal{L}(x'_{2:\lambda}) \leq \cdots \leq \mathcal{L}(x'_{\lambda:\lambda})$
    $r_k^{(t)} = \frac{\lambda - \text{rank}_k}{\sum_{j=1}^{\lambda} j}$
  end for
  //Evaluation and Update
  if $|R_k^{(t)}| \geq W$ **then**
    move the first element in $R_k^{(t)}$
  end if
  $R_k^{(t+1)} = R_k^{(t)} \oplus r_k^{(t)}$
  $p_i^{(t+1)} = \alpha p_i^{(t)} + (1-\alpha) \cdot \bar{R}_i$
  $\epsilon^{(t+1)} = \epsilon_{\min} + (\epsilon_{\max} - \epsilon_{\min}) \cdot e^{-\eta t}$
  $\boldsymbol{m}^{(t+1)} = \sum_{i=1}^{\mu} w_i \delta_{i:\lambda}$
  $\mathbf{p}^{(t+1)} = (1-c_c)\mathbf{p}^{(t)} + \sqrt{c_c(2-c_c)\mu_{eff}}\, \frac{\mathbf{m}^{(t+1)} - \mathbf{m}^{(t)}}{\sigma^{(t)}}$
  $\mathbf{C}^{(t+1)} = (1-c_1)\mathbf{C}^{(t)} + c_1 \mathbf{p}^{(t+1)} \mathbf{p}^{(t+1)^{\mathsf{T}}}$
end while

Generate sparse perturbation: At each iteration, based on the λ subspaces selected for sampling, use the CMA-ES algorithm to generate λ adversarial perturbations δ, which are projected into the sampled subspaces $B_i \subset \mathbb{R}^{3\times m\times m}$ to obtain sparse perturbations δ_{B_i}. In addition, we need to constrain the sparse perturbation with $\| \delta_{B_i} \|_p \leq \epsilon$

and add δ_{B_i} to the input image x to obtain adversarial examples x' with $\| x + \delta_{B_i} \|_p \in [0,1]$.

Objective function value evaluation: We use $C\&W$ loss to calculate the corresponding objective function value $\mathcal{L}(x_i')$ and sort it $\mathcal{L}(x'_{1:\lambda}) \leq \mathcal{L}(x'_{2:\lambda}) \leq \cdots \leq \mathcal{L}(x'_{\lambda:\lambda})$. This can obtain the top μ candidate solutions that have the greatest impact on the classifier decision of the target model, and continue to perform perturbation optimization in the next iteration.

Evaluate and update active subspaces: We measure the sampling subspace B_i based on the ranking of the objective function loss $\mathcal{L}(x)$. We set up a reward mechanism for the sampling subspaces, believing that the subspaces ranked first are more active and have a higher probability of obtaining high-quality adversarial examples. Therefore, we give different rewards r_i for the sampling subspaces, update the subspace historical reward queue $R_i^{(t)}$ and adjust the sampling probability $p_i^{(t)}$, and finally update the greedy coefficient $\epsilon^{(t)}$.

Update evolutionary strategy parameters: Select the top μ solutions based on the ranking of the objective function loss, perform weighted maximum likelihood estimation to update the distribution mean of the samples m^t, as well as the evolutionary path p^t and covariance matrix C^t. By adjusting these parameters, the probability of generating the optimal solution increases.

4 Experiment

In this section, we will fully validate the effectiveness of the AS-ES method. We conduct experiments on MNIST, CIFAR10 and ImageNet64 and compare AS-ES method with several state-of-the-art(SOTA) global and sparse adversarial attack methods. We aim to achieve efficient adversarial attacks and generate high-quality AEs. In addition, we conducted ablation experiments to validate the effectiveness of the AS-ES method mechanism and select the most efficient experimental parameters.

We use attack success rate (ASR), average query count (AQ), and perturbations norm ℓ_p to assess the attack performance of the algorithm. Generally speaking, larger perturbations can improve ASR, but at the same time, they can also lead to the low quality of the AEs. Therefore, we use the ρ metric to measure the ratio of the perturbation ℓ_2 to the ASR, with smaller values indicating better attack performance.

$$\rho = \frac{average\ \ell_2}{ASR} \tag{22}$$

4.1 Comparative Experiment

We compare AS-ES with several SOTA adversarial attacks, including the black-box global attack methods ZOO [9], AutoZOOM [10], the black-box sparse attack methods Square Attack [7], CornerSearch [21], One-Pixel [18]. All parameter settings of these comparison methods are consistent with the original text. We set the side length m of the active subspace to 10% of the input image.

MNIST. The MNIST dataset consists of handwritten digit images and numerical labels ranging from 0 to 9. Each image is a 28×28 pixel grayscale handwritten digit image, with a total of 10 categories. We randomly select 1000 correctly classified test set images for the attack, set the maximum query count Q_{max} to 1000, and select the pre-trained VGG16 model as the target model.

Table 1. Attack performance comparison on MNIST

Method	ASR	AQ	average ℓ_0	average ℓ_1	average ℓ_2	ρ with 50 queries
ZOO	0.001	832	784	2.458	48.764	—
AutoZOOM	0.063	586	784	1.653	29.451	4202.85
Square	0.412	301	274.47	32.178	2.977	10.904
CornerSearch	0.045	657	**1.089**	1.078	1.026	—
AS-ES(ours)	**0.585**	**164**	4	**1.043**	**0.925**	**3.069**

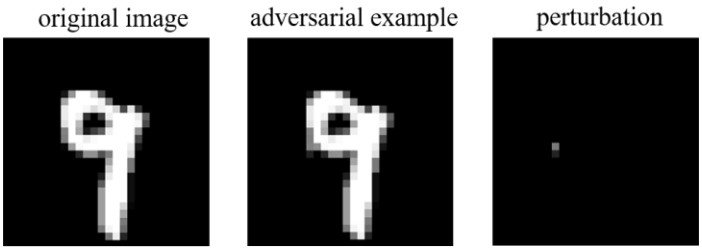

Fig. 1. An example of the original image, adversarial example, and the perturbation on MNIST. It shows that the perturbation is rather sparse.

Evaluation: Table 1 presents the experimental results comparing the method with the AS-ES algorithm on the MNIST dataset. In black-box attacks, our AS-ES method achieved attack performance comparable to white-box attacks, while the success rate of other compared methods was less than 0.5. The AS-ES attack achieves good sparsity, with the lowest perturbation norm compared to other attack methods, and can generate high-quality adversarial examples. The AS-ES method has the lowest ρ value, achieving the best attack performance. Figure 1 shows the visualization of the original image and adversarial examples on the MNIST dataset, indicating that our method generates small and imperceptible perturbations.

CIFAR10. The CIFAR-10 dataset contains a total of 60000 samples, each of which is a 32×32 pixel RGB color image, divided into 10 categories. We randomly select 1000 correctly classified testset images for our experiments, and normalize the input space into $[0,1]^{dim(x_{ori})}$ by $x_{ori}/255$, and set the maximum number of queries for the attack Q_{max} to 1000. We choose the pre-trained VGG16 model as the threat model for the attack and set the active subspace size to m=3.

Table 2. Attack performance comparison on CIFAR10

Method	ASR	AQ	average ℓ_0	average ℓ_1	average ℓ_2	ρ with 50 queries
ZOO	0.092	758	3072	4.913	56.748	14183.75
AutoZOOM	0.316	625	3072	3.704	32.152	1284.81
Square	0.784	417	1758.57	29.067	2.974	13.123
CornerSearch	0.462	854	3.275	2.025	1.171	—
One-Pixel	0.002	897	**3**	1.902	0.945	—
AS-ES(ours)	**0.818**	**168**	27	**1.773**	**0.913**	**1.892**

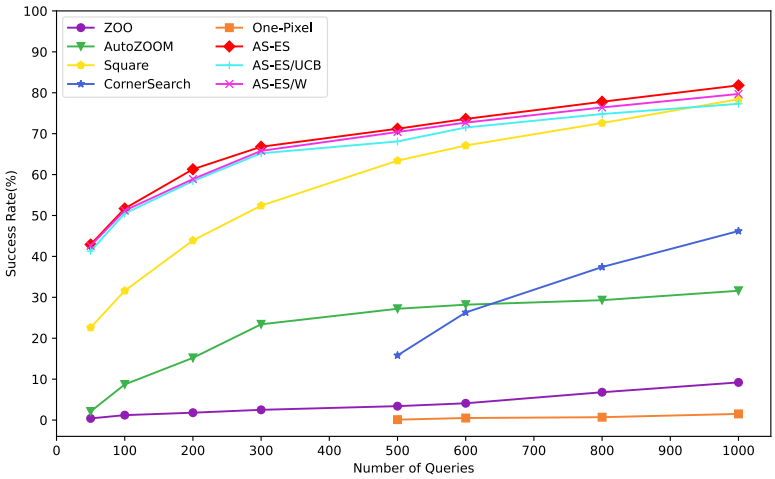

Fig. 2. The attack success rate varies with the query number on CIFAR10. It can be observed that AS-ES achieves a high success rate with the number of queries increasing from 50 to 1000.

Evaluation: Table 2 shows the comparison of attack performance between the comparison method and our AS-ES algorithm in CIFAR10 dataset. In the black-box attacks, our AS-ES algorithm has a higher attack success rate than several state-of-the-art attack methods. The ZOO attack has a low success rate and generates large perturbations. The One-Pixel attack achieves sparsity by only changing one pixel, but the success rate is extremely low. Square attacks generate perturbations are easily noticeable. Our AS-ES attack requires only a small number of queries to achieve a high success rate, and has performance comparable to white-box attacks. From Figure 2, it can be seen that as the number of queries increases, the success rate of the AS-ES method is much higher than other state-of-the-art comparison methods, proving that the AS-ES method has high attack performance. In addition, Our AS-ES algorithm has the lowest ρ value compared

to other state-of-the-art attack methods. Figure 3 shows the visualization of sparse and imperceptible perturbation and adversarial samples generated on the CIFAR10 dataset.

Fig. 3. An example of the original image, adversarial example, and the perturbation on CIFAR10. It shows that the perturbation is rather sparse.

ImageNet64. The ImageNet64 dataset is a subset of ImageNet, where the validation set contains a total of 10,000 images, each of which is an RGB color image of 64×64 size, divided into a total of 200 categories. For evaluation, we randomly select 1000 correctly classified images from the validation set and normalize them. We set the maximum query count Q_{max} for the attack to 500 and select the pre-trained VGG16 model as the target model for the attack.

Table 3. Attack performance comparison on ImageNet64

Method	ASR	AQ	average ℓ_0	average ℓ_1	average ℓ_2	ρ with 50 queries
ZOO	0.005	914	12288	11.318	65.472	—
AutoZOOM	0.347	896	12288	7.762	43.823	504.581
Square	0.614	386	8923.097	189.115	4.972	12.634
One-Pixel	0.011	852	**3**	**1.795**	1.102	—
AS-ES(ours)	**0.763**	**154**	108	5.413	**0.948**	**1.706**

Evaluation: From Table 3, it can be seen that compared with other comparative attack methods, our AS-ES method achieved a higher success rate in black-box attacks, and the average number of queries is only 154, and the generated perturbations are sparse and imperceptible. Among them, the ρ value is the lowest and has the lowest perturbation ℓ_2 norm size. Our AS-ES method has comparable attack performance to white-box attacks, achieving a high success rate and query efficiency, and generating high-quality adversarial examples. Figure 4 shows the results of the attack on the ImageNet64 dataset, and it can be observed from the experiment that the perturbation generated by the AS-ES attack is small and imperceptible.

Fig. 4. An example of the original image, adversarial example, and the perturbation on ImageNet64. It shows that the perturbation is rather sparse.

4.2 Ablation Study

We evaluated the active subspace side length m in the AS-ES method and the mechanism of the sliding window introduced when using the MAB method to locate the active subspace. By setting different subspace sizes m and sliding window sizes w for experiments, we can select the settings of the effective experimental parameters.

Table 4. Ablation Study of Active Subspace Size.

DataSet	size m	ASR	AQ	average ℓ_0	ρ with 50 queries
CIFAR10	1	0.411	187	3	2.988
	2	0.564	182	9	1.973
	3	**0.818**	**168**	**27**	**1.892**
	4	0.824	150	48	2.062
	5	0.881	126	75	2.221
ImageNet64	3	0.575	168	27	3.178
	6	**0.763**	**154**	**108**	**1.706**
	9	0.774	152	243	4.153
	12	0.812	138	432	6.347
	15	0.843	133	675	7.206

Active Subspace Size *m*. Different active subspace sizes imply different sparsity of the generated perturbations. In general, when setting large active subspace side m, this can get the higher ASR but the lower the quality of the generated adversarial examples. Therefore, in practical attack scenarios, setting a reasonable size of active subspace side length m can not only achieve a high ASR, but also obtain sparse and imperceptible adversarial examples.

In Table 4, we set different sizes of subspace side length m values based on different datasets. As the value of m increases, ASR gradually increases, and the average perturbation ℓ_0 norm also increases. The average queries (AQ) shows a decreasing region, which is consistent with intuition. However, it can obtain the higher ρ values, which

means that the attack performance gradually decreases. In our work, we set $m = 3$ for the CIFAR10 dataset and $m = 6$ for the ImageNet64 dataset, which not only can obtain a high ASR and low number of queries, but also has the best attack performance. When m is set too low, it can also lead to ASR being too low, which is not conducive to finding suitable adversarial examples.

Sliding Windows Size *W*. We introduce a sliding window mechanism in the MAB algorithm to effectively solve the dynamic adaptation problem in non-stationary environments. Retaining only the most recent w historical data and ignoring historical old data that exceeds the sliding window size, which can avoid outdated reward information misleading current decisions.

Table 5. Ablation Study of Different Sliding Windows Size.

Window Size W	ASR	AQ	average ℓ_0	ρ with 50 queries
10	0.769	171	0.942	1.936
20	0.755	176	0.949	1.981
30	**0.818**	**168**	**0.913**	**1.892**
50	0.733	184	0.953	1.955
100	0.797	189	0.968	1.978

From Table 5, it can be seen that setting different sliding window sizes w will affect the performance of the attack. We sets the sliding window size to $w = 30$, which can achieve high attack success rate with fewer queries and ensure that the generated adversarial examples have high quality.

In addition, we evaluated the effectiveness of using UCB (Upper Confidence Bound) and sliding window mechanism in the AS-ES method. As shown in Figure 2, AS-ES/UCB represents the attack result where the UCB mechanism is not introduced for sampling, but only random sampling is performed. AS-ES/W does not introduce a sliding window mechanism, but instead retain all historical data. From the experimental results, it can be seen that as the number of queries increases, compared to the AS-ES method, the AS-ES/UCB method and the AS-ES/W method have slightly lower ASR, but the generated perturbation norm is greater than AS-ES, this means that the quality of adversarial examples generated is lower. Therefore, the upper limit of the sampling confidence interval UCB and the sliding window mechanism have achieved better attack performance in terms of attack success rate and adversarial example quality.

5 Conclusion

In this paper, we propose an adversarial example generation method based on AS-ES, which guides black-box attacks to conduct perturbation search in an active subspace. In addition, we formulate a subspace sampling evaluation mechanism and use the

CMA-ES algorithm for perturbation search, which dynamically learns sensitive active subspaces in the input image while evaluating the objective function for the perturbation. The experimental results show that under black-box attacks and query restrictions, the attack performance of AS-ES method is significantly better than the existing SOTA attack methods. It can reduce the number of queries in perturbed search while ensuring the success of the attack, and generate sparse and imperceptible high-quality adversarial examples. This method has greater utility in real-world application scenarios while balancing the goals of query efficiency and imperceptibility.

In future research, we will attempt to find more effective active subspace localization methods that better balance the query efficiency, attack success rate, and imperceptibility of black-box attacks. In addition, we can apply the method to defense systems and test different types of defense models, thus improving the robustness of the models and verifying their effectiveness.

References

1. 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.
2. Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu. Convolutional neural networks for speech recognition. IEEE/ACM Transactions on audio, speech, and language processing, 22(10):1533–1545, 2014.
3. Boris S Kerner. Failure of classical traffic flow theories: Stochastic highway capacity and automatic driving. Physica A: Statistical Mechanics and its Applications, 450:700–747, 2016.
4. Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In Proceedings of the 2016 acm sigsac conference on computer and communications security, pages 1528–1540, 2016.
5. C Szegedy. Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199, 2013.
6. Fengfan Zhou, Qianyu Zhou, Xiangtai Li, Xuequan Lu, Lizhuang Ma, and Hefei Ling. Adversarial attacks on both face recognition and face anti-spoofing models. arXiv preprint arXiv:2405.16940, 2024.
7. Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In European conference on computer vision, pages 484–501. Springer, 2020.
8. Aleksander Mkadry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. stat, 1050(9), 2017.
9. Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM workshop on artificial intelligence and security, pages 15–26, 2017.
10. Chun-Chen Tu, Paishun Ting, Pin-Yu Chen, Sijia Liu, Huan Zhang, Jinfeng Yi, Cho-Jui Hsieh, and Shin-Ming Cheng. Autozoom: Autoencoder-based zeroth order optimization method for attacking black-box neural networks. In Proceedings of the AAAI conference on artificial intelligence, volume 33, pages 742–749, 2019.

11. HanQin Cai, Yuchen Lou, Daniel McKenzie, and Wotao Yin. A zeroth-order block coordinate descent algorithm for huge-scale black-box optimization. In International Conference on Machine Learning, pages 1193–1203. PMLR, 2021.
12. Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16, pages 35–50. Springer, 2020.
13. Francesco Croce, Maksym Andriushchenko, Naman D Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: a versatile framework for query-efficient sparse black-box adversarial attacks. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 6437–6445, 2022.
14. Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. arXiv preprint arXiv:1807.07978, 2018.
15. Viet Quoc Vo, Ehsan Abbasnejad, and Damith C Ranasinghe. Query efficient decision based sparse attacks against black-box deep learning models. arXiv preprint arXiv:2202.00091, 2022.
16. Zhaoyu Chen, Bo Li, Shuang Wu, Shouhong Ding, and Wenqiang Zhang. Query-efficient decision-based black-box patch attack. IEEE Transactions on Information Forensics and Security, 2023.
17. Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In 2016 IEEE European symposium on security and privacy (EuroS&P), pages 372–387. IEEE, 2016.
18. Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. IEEE Transactions on Evolutionary Computation, 23(5):828–841, 2019.
19. Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. Advances in Neural Information Processing Systems, 33:11226–11236, 2020.
20. Zeyu Dai, Shengcai Liu, Qing Li, and Ke Tang. Saliency attack: towards imperceptible black-box adversarial attack. ACM Transactions on Intelligent Systems and Technology, 14(3):1–20, 2023.
21. Matthias Hein and Francesco Croce. Sparse and imperceivable adversarial attacks. In 2019 IEEE International Conference on Computer Vision (ICCV). IEEE, 2019.
22. 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.