Black-box Adversarial Attacks with Limited Queries and Information

Kemasaram Nithin, Ganta Srujan, K R Eashwar Sai

The source code for the experiments and implementations described in this report can be found at repo.

1 Abstract

This research investigates the vulnerability of neural network-based classifiers to adversarial examples in restrictive black-box settings, where attackers have limited query access and information. Three realistic threat models are defined: query-limited, partial-information, and label-only settings. Novel attack algorithms are developed to generate adversarial examples under these constraints, demonstrating effectiveness against an ImageNet classifier and a commercial system, the Google Cloud Vision API. The proposed methods significantly improve query efficiency and succeed in targeted misclassification, even with minimal information, highlighting the persistent vulnerability of machine learning systems in real-world scenarios.

2 Introduction

Neural network-based image classifiers are prone to adversarial examples—inputs with subtle perturbations that cause misclassification. While white-box attacks leverage full model access, real-world systems often operate under stricter black-box conditions, where attackers can only query the model. This study introduces three constrained black-box threat models:

1. **Query-limited setting**: Attackers have a restricted number of queries due to time, cost, or rate limits.
2. **Partial-information setting**: Only top-k class probabilities or scores are available, often non-normalized.
3. **Label-only setting**: Only the top-k sorted labels are provided, without scores.

These models reflect practical constraints in commercial systems like the Clarifai NSFW API, Google Cloud Vision API, and Google Photos. The research aims to develop query-efficient algorithms for generating targeted adversarial examples under these restrictions, addressing limitations of prior methods that require millions of queries or full probability outputs. The study evaluates the proposed attacks on an ImageNet classifier and demonstrates a real-world attack on the Google Cloud Vision API.

3 Methodology

The research employs a combination of mathematical techniques and algorithms to generate adversarial examples under the proposed threat models. The goal is to find an adversarial input x_{adv} such that $\|x_{\text{adv}} - x\|_\infty < \epsilon$ and x_{adv} is classified as the target class y_{adv}, given an original input x, perturbation bound ϵ, and black-box access to the classifier's output probabilities or labels. Below, we detail the key mathematical foundations and algorithms used for each threat model.

3.1 Notation

- **Projection Operator**: $\Pi_{[x-\epsilon,x+\epsilon]}(x')$ denotes the ℓ_∞-norm projection of x' onto an ϵ-ball around x, abbreviated as $\Pi_\epsilon(x')$ or $\text{CLIP}(x', x-\epsilon, x+\epsilon)$ in pseudocode.
- **Rank Function**: $\text{rank}(y|x)$ returns the smallest k such that class y is among the top-k classes for input x.
- **Distributions**: $\mathcal{N}$ and $\mathcal{U}$ represent normal and uniform distributions, respectively.

3.2 Query-Limited Setting

In the query-limited setting, the attacker has a query budget L and seeks to estimate the gradient of the classifier's output probability $P(y|x)$ efficiently. The approach uses **Natural Evolutionary Strategies (NES)**, inspired by Wierstra et al. (2014), to estimate gradients, followed by **Projected Gradient Descent (PGD)** to generate adversarial examples.

3.2.1 Natural Evolutionary Strategies (NES)

NES is a derivative-free optimization method that maximizes the expected value of a loss function $F(x)$ under a search distribution $\pi(\theta|x)$. For a classifier's output probability $P(y|x)$, the gradient is estimated as follows:

$$\mathbb{E}_{\pi(\theta|x)}[F(\theta)] = \int F(\theta)\pi(\theta|x)\, d\theta$$

$$\nabla_x \mathbb{E}_{\pi(\theta|x)}[F(\theta)] = \mathbb{E}_{\pi(\theta|x)}[F(\theta)\nabla_x \log \pi(\theta|x)]$$

The search distribution is chosen as Gaussian noise around the current image: $\theta = x + \sigma\delta$, where $\delta \sim \mathcal{N}(0, I)$. Using antithetic sampling (Salimans et al., 2017), $n/2$ Gaussian noise vectors δ_i are sampled, and their negatives are used to form a population of size n. The gradient estimate is:

$$\nabla \mathbb{E}[F(\theta)] \approx \frac{1}{\sigma n} \sum_{i=1}^{n} \delta_i F(x + \sigma\delta_i)$$

This is implemented in **Algorithm 1 (NES Gradient Estimate)**:

- **Input**: Classifier $P(y|x)$, image x.
- **Parameters**: Search variance σ, number of samples n, image dimensionality N.

- For each iteration, sample $u_i \sim \mathcal{N}(0, I_{N\times N})$, compute $g \leftarrow g + P(y|x+\sigma u_i)\cdot u_i - P(y|x-\sigma u_i)\cdot u_i$, and return $\frac{1}{2n\sigma}g$.

The NES estimate is equivalent to a finite-difference method over random Gaussian bases, with theoretical guarantees from the Johnson-Lindenstrauss Theorem ensuring the estimated gradient norm $\|\hat{\nabla}\|^2$ approximates the true gradient norm $\|\nabla\|^2$:

$$\mathbb{P}\left\{(1-\delta)\|\nabla\|^2 \leq \|\hat{\nabla}\|^2 \leq (1+\delta)\|\nabla\|^2\right\} \geq 1-2p$$

where $0 < \delta < 1$, and the number of samples $N = O(-\delta^{-2}\log p)$.

3.2.2 Query-Limited Attack

The attack uses the NES gradient estimate in a PGD update:

$$x^{(t)} = \Pi_{[x_0-\epsilon, x_0+\epsilon]}\left(x^{(t-1)} - \eta\cdot\operatorname{sign}(g_t)\right)$$

where g_t is the estimated gradient, η is the step size, and $\frac{L}{N}$ PGD steps are performed with N queries per gradient estimate.

3.3 Partial-Information Setting

Only the top-k class probabilities (or scores) are available in the partial-information setting. The algorithm starts with an image x_0 of the target class y_{adv} and alternates between:

1. Projecting onto decreasing ℓ_∞-balls centered at the original image x.
2. Maximizing the probability of y_{adv}.

3.3.1 Algorithm 2 (Partial Information Attack)

- **Input**: Original image x, target class y_{adv}, classifier $P(y|x)$ returning top-k probabilities.
- **Parameters**: Perturbation bound ϵ_{adv}, initial perturbation ϵ_0, NES parameters (σ, N, n), epsilon decay δ_ϵ, learning rates $\eta_{\max}, \eta_{\min}$.
- **Process**:
  - Initialize x_{adv} as an image of y_{adv}, clipped to $[x-\epsilon_0, x+\epsilon_0]$.
  - While $\epsilon > \epsilon_{\text{adv}}$ or the top class is not y_{adv}:
    - Compute gradient g using NES for $P(y_{\text{adv}}|x_{\text{adv}})$.
    - Perform a backtracking line search to find $\eta \leq \eta_{\max}$ such that y_{adv} remains in the top-k classes after updating $\hat{x}_{\text{adv}} = \text{CLIP}(x_{\text{adv}} - \eta g, x-\epsilon, x+\epsilon)$.
    - If $\eta < \eta_{\min}$, increase ϵ and halve δ_ϵ.
    - Update $x_{\text{adv}} \leftarrow \hat{x}_{\text{adv}}$, decrease $\epsilon \leftarrow \epsilon - \delta_\epsilon$.

The projection step ensures $\epsilon_t = \min \epsilon'$ such that $\text{rank}(y_{\text{adv}}|\Pi_{\epsilon'}(x^{(t-1)})) < k$, maintaining the target class in the top-k.

3.4 Label-Only Setting

In the label-only setting, only the top-k sorted labels are available. A proxy score is defined to substitute for missing probabilities:

$$R(x^{(t)}) = k - \text{rank}(y_{\text{adv}}|x^{(t)})$$

The robustness of the adversarial image to random perturbations is used as a proxy for classification confidence:

$$S(x^{(t)}) = \mathbb{E}_{\delta \sim \mathcal{U}[-\mu,\mu]}[R(x^{(t)} + \delta)]$$

This is approximated via Monte Carlo sampling:

$$\hat{S}(x^{(t)}) = \frac{1}{n} \sum_{i=1}^{n} R(x^{(t)} + \mu\delta_i)$$

The partial-information algorithm is adapted by treating $\hat{S}(x)$ as a proxy for $P(y_{\text{adv}}|x)$, using NES to estimate $\nabla_x \hat{S}(x)$.

4 Related Work

Adversarial attacks have been extensively studied, with early work by Szegedy et al. (2013) and Biggio et al. (2012) establishing the vulnerability of classifiers to adversarial examples. White-box attacks, such as those by Goodfellow et al. (2015) and Carlini & Wagner (2017), assume full model access. Black-box attacks, more relevant to real-world systems, include:

- **Substitute Networks**: Papernot et al. (2016a; 2017) trained substitute models to emulate target classifiers, generating transferable adversarial examples. Liu et al. (2017) used ensemble methods but achieved low transferability (18%
- **Gradient Estimation**: Chen et al. (2017) estimated gradients via pixel-wise finite differences, requiring hundreds of thousands of queries. Narodytska & Kasiviswanathan (2017) used local search, averaging 17,000 queries for CIFAR-10 attacks.
- **Limited Information**: Brendel et al. (2018) explored decision-based attacks, similar to the label-only setting, using boundary-based methods.

Other works, like Xu et al. (2016) and Nguyen et al. (2014), applied evolutionary algorithms but focused on different goals or domains. This study distinguishes itself by addressing targeted attacks under multiple restrictive threat models, improving query efficiency, and targeting complex commercial systems.

5 Experiment

The experiments evaluate the proposed attack algorithms under the three threat models using a pre-trained InceptionV3 ImageNet classifier (78%

- **Dataset**: 1000 randomly chosen ImageNet test images (100 for label-only) with randomly selected target classes.
- **Perturbation Limit**: ℓ_∞-bounded perturbation with $\epsilon = 0.05$.
- **Metrics**: Success rate (percentage of adversarial examples classified as the target class) and median number of queries.
- **Threat Models**:
  - **Query-limited**: Uses NES for gradient estimation, followed by PGD.
  - **Partial-information**: Starts with a target class image, alternates between blending with the original image and maximizing target class likelihood using NES and PGD.
  - **Label-only**: Employs a proxy score based on label ranking robustness to random perturbations, integrated into the partial-information algorithm.
- **Real-world Attack**: A targeted attack on the Google Cloud Vision API, a partial-information classifier with unknown class enumeration and non-probabilistic scores.

Hyperparameters, such as NES search variance ($\sigma = 0.001$), population size ($n = 50$), and learning rate ($\eta = 0.005$), were fixed across all images. The query limit was capped at 10^6 for query-limited attacks, with distributions analyzed.

6 Proposed Methodology and Novel Contributions

In this work, we propose a series of improvements to black-box adversarial attacks in label-only settings by combining three key ideas: **(1) CAM-guided perturbations**, **(2) patch-wise gradient estimation**, and **(3) adaptive temperature-based ϵ scheduling**. We now describe each component in detail.

6.1 CAM-Guided Perturbation Prior

We utilize Class Activation Maps (CAMs) to guide adversarial perturbations towards the most discriminative regions of the input image. Formally, given an input $x \in \mathbb{R}^{C\times H\times W}$ and a classifier $f(\cdot)$, the CAM for a class c provides a spatial map $M_c(x) \in \mathbb{R}^{H\times W}$ indicating pixel importance.

We generate a normalized CAM mask:

$$\text{CAM\_mask}(x) = \frac{M_c(x)}{\max(M_c(x))}$$

This serves as a weighting prior during gradient estimation. During NES (Natural Evolution Strategies) gradient approximation, perturbations are sampled with importance weights according to $\text{CAM\_mask}(x)$, focusing the search on highly influential regions. Focusing perturbations on regions critical to model decision-making is more likely to affect the output with fewer queries, improving efficiency.

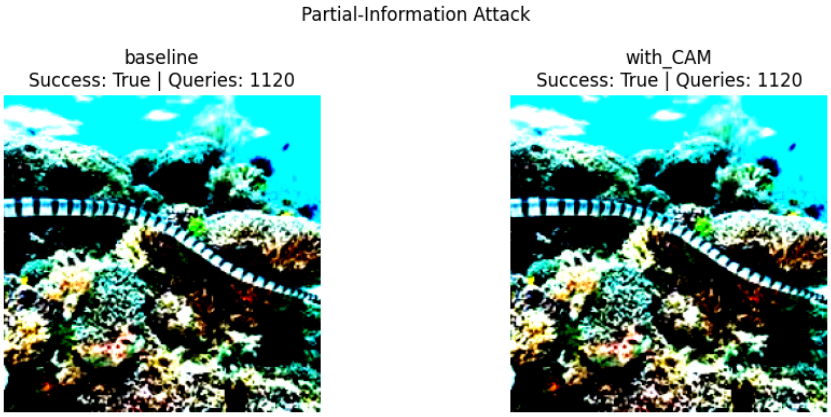

Figure 1: The figure shows the adversarial images generated for the baseline attack and grad_CAM attack

6.2 Patch-Wise Gradient Estimation

Rather than perturbing the entire image at once, we divide the input into non-overlapping patches of size $P \times P$ and perform gradient estimation locally. The full image is partitioned into $\frac{H}{P} \times \frac{W}{P}$ patches.

For each patch (i, j), the estimated gradient is:

$$\hat{g}_{ij} = \frac{1}{n} \sum_{k=1}^{n} \left(f(x + \sigma u_{ij}^{(k)}) - f(x) \right) u_{ij}^{(k)}$$

where $u_{ij}^{(k)}$ is a random perturbation restricted to patch (i, j), σ is the noise scale, and n is the number of NES samples. Restricting perturbations to patches reduces the attack search space, leading to faster convergence, and synergizes with the CAM mask by refining perturbations only in important regions.

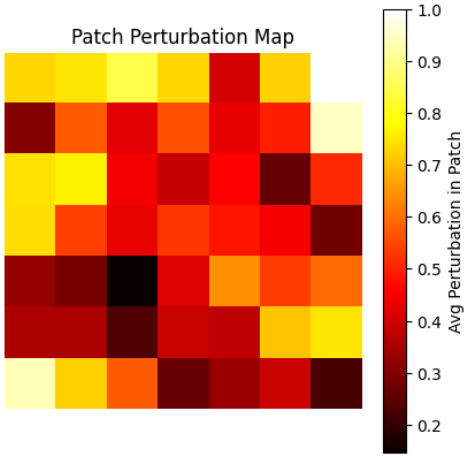

Figure 2: The generated patch perturbation map in the patch wise attack

6.3 Temperature-Based Epsilon Scheduling

To balance attack strength and stealthiness, we adaptively decay the perturbation budget ϵ over iterations using a cosine annealing schedule:

$$\epsilon_t = \epsilon_{\text{final}} + (\epsilon_{\text{init}} - \epsilon_{\text{final}}) \times 0.5 \left(1 + \cos\left(\pi \frac{t}{T}\right)\right)$$

Where t is the current iteration and T is the total number of iterations. Early in the attack, larger ϵ allows aggressive exploration; as the attack progresses, shrinking ϵ enforces fine, precise perturbations, reducing the chance of overshooting and maintaining imperceptibility.

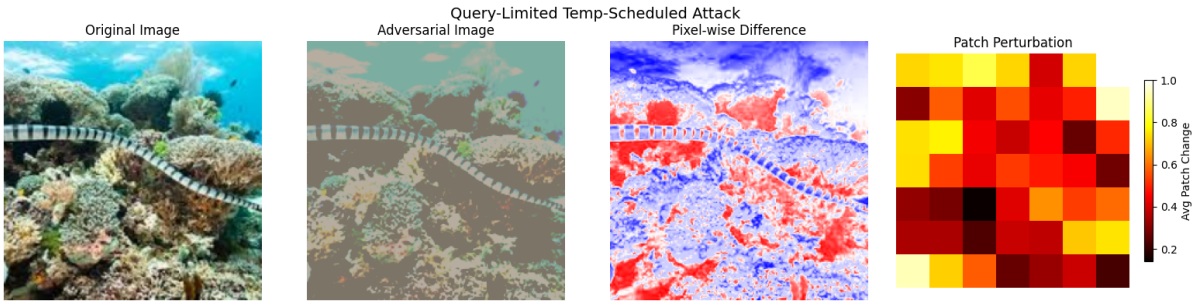

Figure 3: grad_CAM + patch + temp_scheduled epsilon

6.4 Attack Strategy

The final attack integrates all components:

- Use CAM mask to guide NES perturbations.
- Apply patch-wise restricted updates.
- Decay ϵ using temperature scheduling.

The adversarial update step at iteration t becomes:

$$x_{t+1} = \text{Clip}\left(x_t + \eta \cdot \text{sign}(\hat{g}_t),\ x_{\text{orig}},\ \epsilon_t\right)$$

where $\hat{g}_t$ is the CAM-weighted, patchwise NES gradient, η is the step size, and Clip$(\cdot)$ projects the adversarial example within the allowed ℓ_∞ ball of radius ϵ_t around x_{orig}.

Our method attacks the classifier more efficiently by:

1. Prioritizing regions where perturbations are most impactful (via CAMs),
2. Reducing query complexity by focusing perturbation effort locally (via patches),
3. Preventing over-perturbation and fine-tuning adversarial examples over time (via temperature-based ϵ decay).

Overall, this design leads to a highly query-efficient and effective black-box adversarial attack, even under strict label-only access scenarios.

7 Results and Discussion

Our experiments show that the proposed combined attack strategy significantly reduces the number of queries needed to successfully generate adversarial examples compared to baseline methods. Notably, the integration of CAM-guided perturbations, patch-wise gradient estimation, and temperature-based ϵ scheduling yields a more lethal attack in terms of query efficiency, particularly in the label-only and partial-information settings.

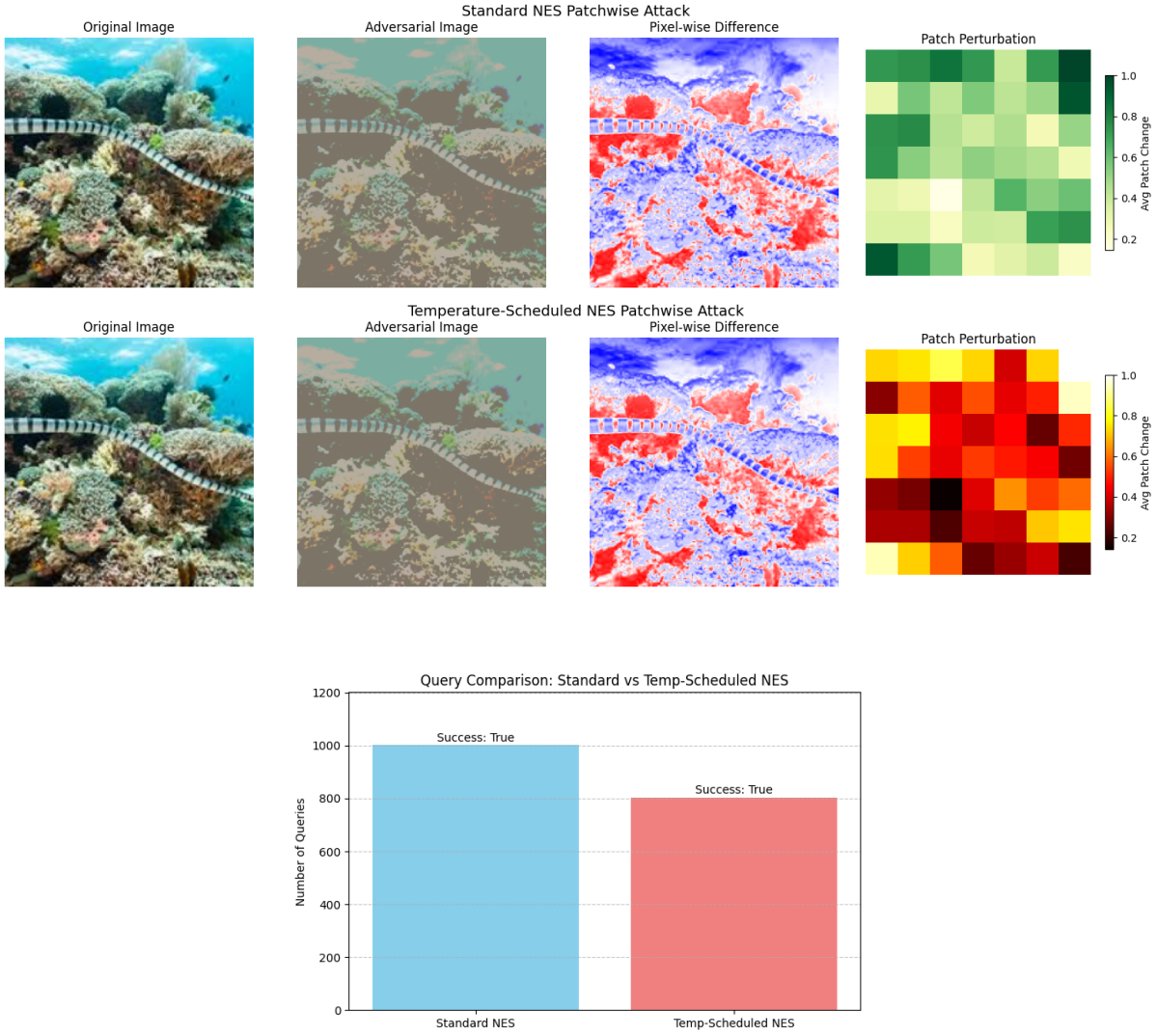

Figure 5: Query comparison

In the **query-limited attack** setting, we observe a modest reduction in queries when employing the combined strategy, while the improvements become more pronounced in harder scenarios. Specifically, under the **partial-information** and **label-only** assumptions, where the adversary must infer gradients from top-k labels or single labels respectively, the combined method achieves a substantial decrease in query counts compared to simple patch-wise or CAM-guided variants.

Interestingly, when employing **only CAM guidance** or **only patch-wise gradient estimation** (without temperature scheduling), the number of queries did not consistently decrease. In some cases, these standalone strategies even led to slightly higher query requirements. A possible explanation stems from the inherent noise and instability

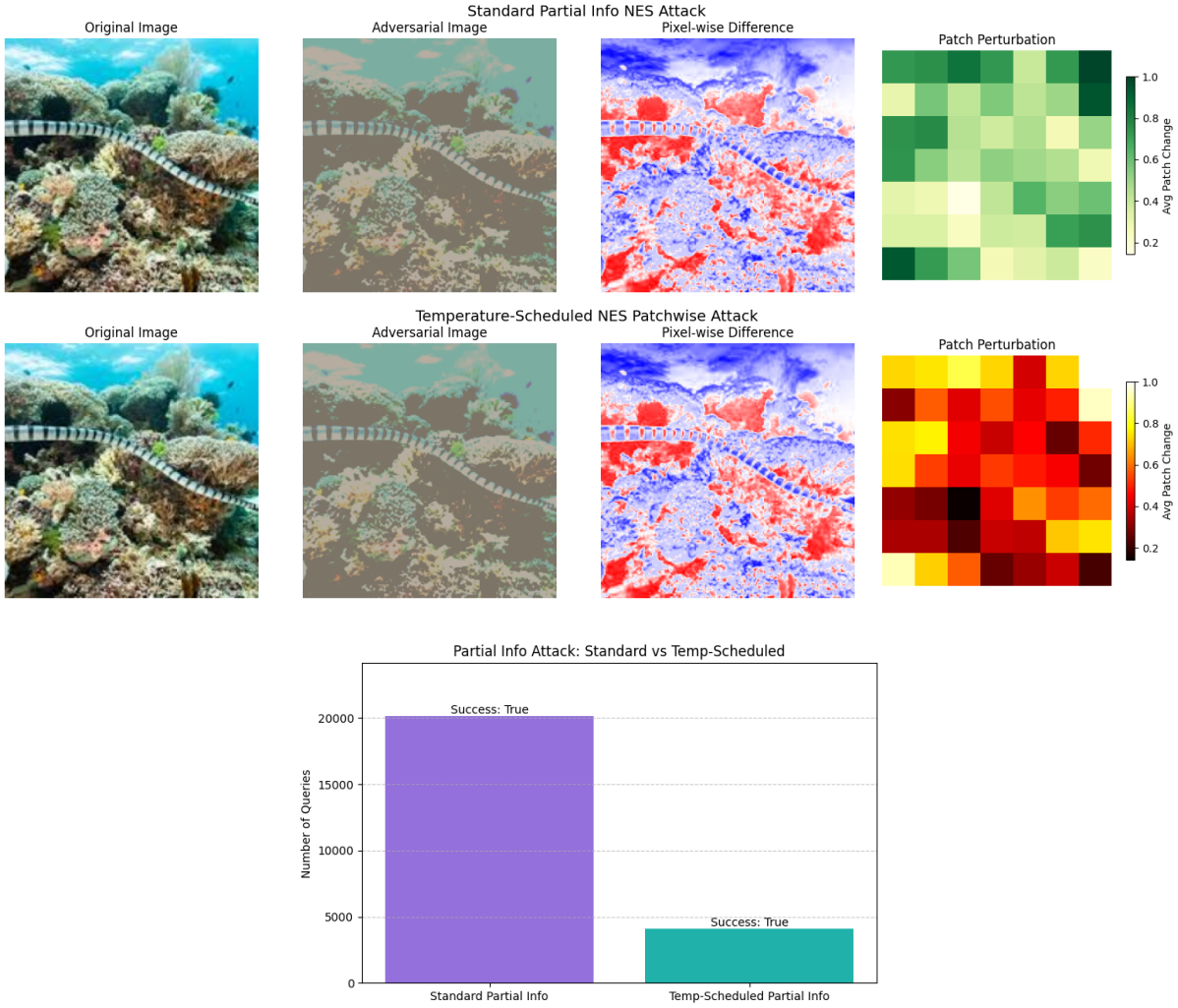

Figure 7: Query comparison in case of partial info setting

introduced by each component in isolation. When only using CAMs, the perturbation space is biased towards salient regions, but without adaptive control over the perturbation budget, the attack risks perturbing the same salient areas repeatedly, causing redundancy and stagnation. Conversely, patch-wise perturbations, while reducing the search space, may lack sufficient global context, causing inefficient exploration when the attack budget ϵ remains static.

Mathematically, if the optimization objective is $J(x)$ (e.g., decreasing the rank of target label), and the perturbation update at step t is Δx_t, then without temperature scheduling,

$$\Delta x_t = \eta \cdot \text{sign}(\hat{g}_t)$$

where $\hat{g}_t$ is the noisy gradient approximation. If $\|\Delta x_t\|_\infty$ is too large and remains constant, it can overshoot the narrow adversarial manifold, resulting in wasted queries without successful misclassification.

However, once **temperature-based ϵ scheduling** is introduced, the perturbation magnitude $\|\Delta x_t\|_\infty \leq \epsilon_t$ dynamically shrinks over time. Early stages with large ϵ_t promote aggressive exploration across patches and CAM-weighted regions, while later stages with smaller ϵ_t enable fine-grained, precision attacks that carefully align with decision boundaries. This dynamic adjustment allows the attack to efficiently transition from

coarse perturbations to delicate refinements, dramatically improving convergence.

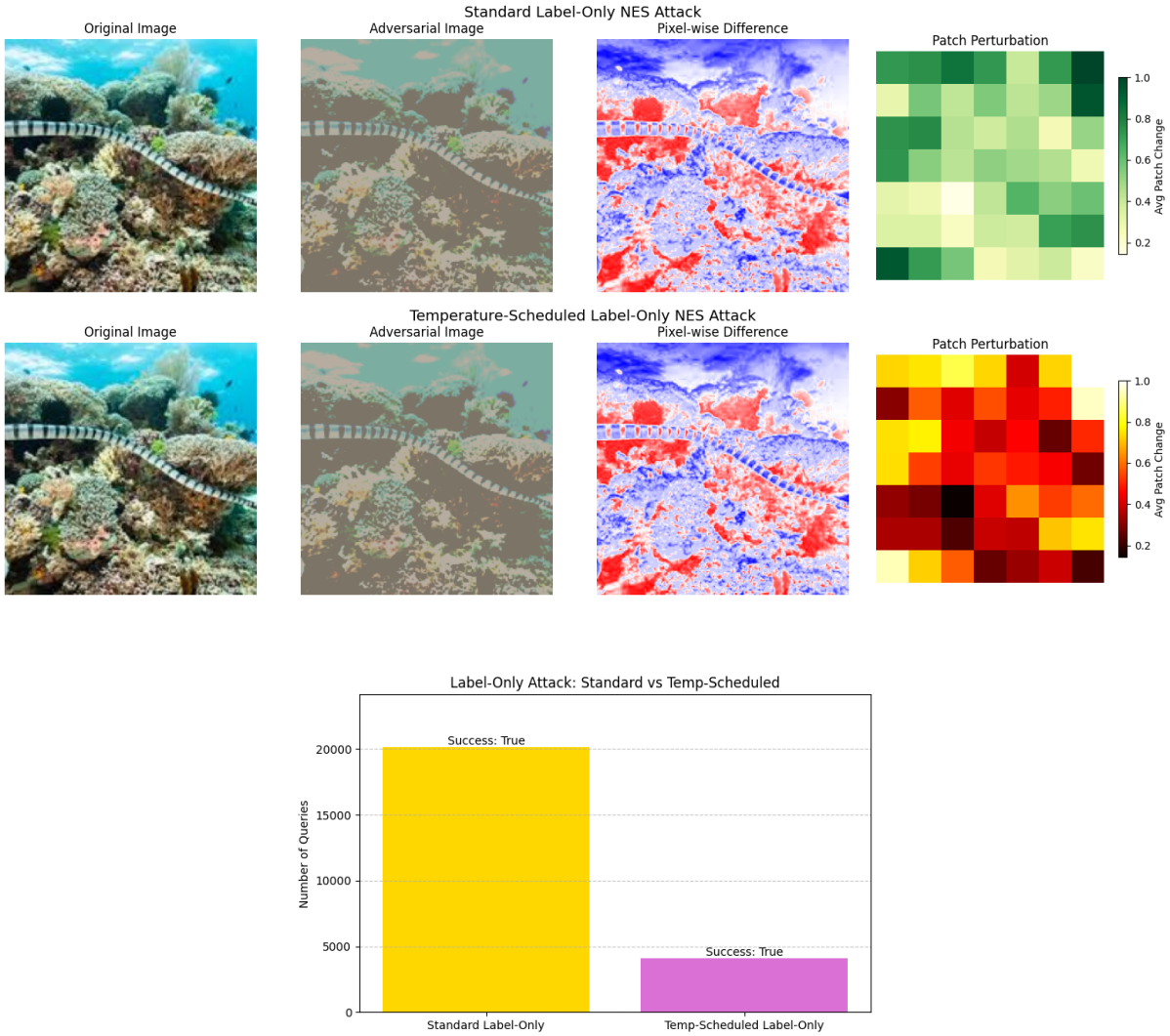

Figure 9: Query comparison in case of label only setting

Moreover, the synergy between CAM-guided priors and patch-wise exploration becomes effective only when the perturbation budget adapts to the stage of the attack. Without temperature scheduling, CAM priors and patch division might cause either overly local exploration or over-saturation of important regions, both of which are suboptimal.

In summary, our empirical results confirm that:

- **CAM-only** and **Patch-only** attacks are insufficient on their own for query efficiency.
- **Combined CAM + Patch** strategy alone does not reliably outperform baselines.
- **Full Combined Strategy (CAM + Patch + Temperature Scheduling)** achieves significant query reductions, particularly in partial and label-only settings.

The experiments were conducted on a custom-curated set of images. The primary objective was to evaluate the effectiveness of the proposed attack strategies under query-limited, partial information, and label-only threat models.

The results consistently supported the hypothesis that the combined use of patch-wise perturbations, Grad-CAM guided weighting, and temperature-scheduled epsilon scaling significantly improves attack efficiency. Notably, this integrated approach led to a substantial reduction in the number of queries required to achieve successful adversarial examples compared to using either patch-wise attacks or Grad-CAM weighting individually. These findings reinforce the importance of adaptive, spatially-aware perturbation strategies in query-constrained adversarial settings.

This demonstrates the necessity of adaptively controlling perturbation strength alongside spatial priors and localized attacks for achieving highly efficient black-box adversarial attacks.