Variance-Reduced Zeroth-Order Optimization Using SVRG-based ZO-AdaMM

EE 782 - Course Project Presentation

Jincy P Janardhanan (24M1075)
Abhinav P (24D0547)
Arnab Mukherjee (24D1597)

Department of Electrical Engineering
IIT Bombay

Introduction

Zeroth-Order (ZO) optimization is essential for black-box settings where gradients of the objective function $f(x)$ are unavailable. Such scenarios appear in:

- adversarial attacks on deep networks,
- hyperparameter tuning,
- scientific simulations,
- non-differentiable or implicit systems.

A major challenge in ZO methods is the **high variance** of gradient estimators, which scales as $\mathcal{O}(d)$ with dimension, leading to slow convergence and high query complexity.

Introduction

Black-box optimization relies on function evaluations without access to gradients. This setting appears in:

- adversarial attacks on deep networks,
- hyperparameter tuning,
- scientific and engineering simulation optimization,
- non-differentiable or implicit systems.

However, ZO gradient estimators suffer from a crucial limitation:

$$\mathrm{Var}(\hat{g}) = \mathcal{O}(d)$$

which grows linearly with dimension d. This leads to:

- slow convergence,
- unstable trajectories,
- high query complexity.

Existing Methods and Their Limitations

1) Basic and Deterministic Methods

Classical ZO-SGD and ZO-GD use randomized finite-difference estimators with fixed learning rates. They are foundational but limited:

- inherit full variance of order $\mathcal{O}(d)$,
- slow convergence in practice,
- deterministic coordinate-descent variants reduce noise but require $\mathcal{O}(d)$ queries per iteration,
- impractical for modern high-dimensional settings (e.g., image-based adversarial attacks).

Existing Methods and Their Limitations

2) Adaptive ZO Methods

Optimizers like ZO-Adam and ZO-AdaMM improve empirical stability through:

- momentum smoothing,
- per-coordinate learning-rate scaling.

They are widely used in black-box adversarial attacks because they handle non-convex, ill-conditioned landscapes better than basic ZO schemes.

Limitation: They still operate directly on noisy finite-difference gradients, so estimator variance remains fundamentally:

$$\mathcal{O}(d)$$

yielding limited convergence improvements.

Existing Methods and Their Limitations

3) Variance-Reduction Methods

Classical VR algorithms (SVRG, SARAH, SPIDER) offer strong guarantees in first-order optimization.

ZO counterparts (ZO-SVRG, ZO-SARAH, ZO-SPIDER) mirror these ideas, but face major issues in black-box settings:

- recursive VR methods accumulate ZO noise across iterations,
- instability becomes severe in high-dimensional problems,
- reliance on fixed global learning rates makes them sensitive to ill-conditioning,
- incompatible with adaptive scaling used in practical ZO attacks.

Consequently: Despite theoretical appeal, existing ZO-VR methods have not become viable for large-scale black-box adversarial tasks.

Motivation

AdaMM offers:

- per-coordinate learning-rate scaling,
- momentum-based smoothing,
- stable updates under heterogeneous curvature and noisy ZO gradients.

SVRG contributes:

- periodic reference-gradient surrogates,
- prevention of noise accumulation,
- alleviation of the $\mathcal{O}(d)$ variance barrier,
- no recursive propagation of estimator noise.

Together, AdaMM + SVRG form a structure naturally suited for high-dimensional adaptive ZO optimization.

Benefits of the Combined Approach

The hybrid ZO-AdaMM-SVRG method:

- reduces gradient-estimator variance without recursive noise amplification,
- preserves adaptive per-coordinate scaling (stability),
- improves convergence speed under realistic query budgets,
- remains robust in high-dimensional, non-convex black-box settings.

These properties address the core weaknesses in both classical ZO optimizers and existing ZO-VR techniques.

Research Gap

A gap exists between theoretical ZO-VR methods and those used in practical black-box adversarial attacks.

Modern high-dimensional attacks rely on:

- NES-style gradient estimators,
- bandit-based priors,
- adaptive ZO optimizers (e.g., ZO-AdaMM),

while recursive VR methods remain unstable under noisy ZO gradients and strict query constraints.

SVRG provides a variance-reduction framework that avoids recursive error accumulation and aligns structurally with adaptive optimizers.

Integrating SVRG with AdaMM yields a variance-efficient and empirically stable ZO method, a combination not achieved by prior ZO-VR approaches.

Literature Review

Zeroth-Order Optimization. Classical derivative-free methods such as Nelder–Mead and CMA-ES laid early foundations, but scale poorly in high dimensions.

Randomized smoothing and NES-style estimators popularized stochastic directional derivatives as practical tools for ZO learning. These enabled black-box attacks and high-dimensional optimization via:

- Gaussian smoothing,
- two-point estimators,
- NES and antithetic sampling,
- structured or orthogonal directions.

These techniques improve empirical efficiency but do not eliminate the $\mathcal{O}(d)$ variance bottleneck.

Literature Review

Adaptive ZO Methods. ZO-Adam, ZO-AdaGrad, and ZO-AdaMM introduced momentum and per-coordinate learning-rate scaling, greatly improving stability in non-convex black-box tasks. However, they still rely on noisy finite-difference gradients and do not fundamentally reduce variance.

Variance-Reduction Methods. First-order VR algorithms (SVRG, SARAH, SPIDER) offer strong theoretical guarantees. ZO variants (ZO-SVRG, ZO-SARAH, ZO-SPIDER) mimic these ideas, but:

- recursive VR accumulates ZO noise,
- often unstable in high dimensions,
- sensitive to fixed global learning rates,
- incompatible with adaptive scaling needed in practice.

These limitations motivate a stable, adaptive, variance-reduced ZO method.

Literature Review

Recent Advances in ZO Estimation.

Several strategies aim to reduce variance more directly:

- **Two-point estimators** Improve constant factors but not the $\mathcal{O}(d)$ variance dependence.
- **Orthogonal / structured direction sampling** Provides better directional coverage but limited asymptotic gains.
- **Antithetic sampling** Reduces noise through paired perturbations.
- **Hybrid ZO approaches with learned priors or models** Empirically reduce noise but remain brittle under strict query constraints.

The literature reveals two main threads:

- adaptive ZO methods: stable but variance-limited,
- ZO-VR methods: theoretically strong but unstable in practice.

No existing method achieves both adaptive stability and robust variance reduction.

Proposed Method: ZO-AdaMM-SVRG

We introduce **ZO-AdaMM-SVRG**, a hybrid optimizer that combines:

- **Adaptive updates (AdaMM)** Momentum + per-coordinate learning-rate scaling for stability under noisy ZO gradients.
- **Variance Reduction (SVRG)** Periodic reference-gradient correction to suppress stochastic noise without recursive accumulation.

Key Idea: Use SVRG-style control variates to reduce estimator variance, while AdaMM ensures stable, adaptive updates.

Zeroth-Order Gradient Estimation

We use **Gaussian smoothing** to approximate the gradient of a black-box function $f(x)$ using only function evaluations.

For a random direction $u \sim \mathcal{N}(0, I_d)$ normalized to unit length, the two-point estimator is:

$$g_\mu(x; u) = \frac{d}{2\mu} \big(f(x + \mu u) - f(x - \mu u)\big)\, u.$$

To reduce noise, we average $\mathbf{q}$ such estimates at each inner iteration:

$$\hat{g}_t = \frac{d}{q} \sum_{i=1}^{q} \frac{f(x_t + \mu u_{t,i}) - f(x_t - \mu u_{t,i})}{2\mu}\, u_{t,i}.$$

This provides a smoothed, direction-averaged approximation to the true gradient, enabling optimization without derivatives.

SVRG-Style Variance Reduction i

To suppress stochastic noise in ZO gradient estimates, we use **SVRG control variates**.

At the start of each epoch, compute a **reference gradient** using a large batch of Q directions:

$$\tilde{g}_s = \frac{d}{Q} \sum_{i=1}^{Q} \frac{f(x_{s,0} + \mu u_i) - f(x_{s,0} - \mu u_i)}{2\mu} u_i.$$

For each inner iteration, compute:

$$\hat{g}_t = \text{ZO gradient at } x_{s,t}, \qquad \hat{g}_t^{\text{ref}} = \text{ZO gradient at } x_{s,0}$$

using the **same directions** $u_{t,i}$.

SVRG-Style Variance Reduction ii

The **variance-reduced estimator** is:

$$g_{s,t}^{\mathrm{VR}} = \hat{g}_t - \hat{g}_t^{\mathrm{ref}} + \tilde{g}_s.$$

This structure:

- maintains unbiasedness,
- cancels shared noise via control variates,
- avoids recursive noise accumulation,
- stabilizes ZO gradients in high dimensions.

Adaptive Moment Estimation

To stabilize updates under noisy ZO gradients, we adopt the **AdaMM** structure with AMSGrad correction.

Given the variance-reduced gradient $g_{s,t}^{\mathrm{VR}}$, the moments are updated as:

$$m_{s,t+1} = \beta_1 m_{s,t} + (1-\beta_1) g_{s,t}^{\mathrm{VR}},$$

$$v_{s,t+1} = \beta_2 v_{s,t} + (1-\beta_2) \big(g_{s,t}^{\mathrm{VR}}\big)^2.$$

AMSGrad correction:

$$\hat{v}_{s,t+1} = \max\big(\hat{v}_{s,t},\, v_{s,t+1}\big).$$

Adaptive parameter update:

$$x_{s,t+1} = x_{s,t} - \alpha_{s,t} \frac{m_{s,t+1}}{\sqrt{\hat{v}_{s,t+1}} + \varepsilon}.$$

Algorithm Structure: ZO-AdaMM-SVRG i

The method follows an **outer–inner loop** structure.

Outer Loop (Epochs):

- Start from point $x_{s,0}$.
- Compute a high-accuracy **reference gradient** $\tilde{g}_s$ using a large batch of Q directions.
- Reset first and second moments.

Inner Loop (Iterations):

- Sample q random directions.
- Compute:
  - ZO gradient at current point $\hat{g}_t$,
  - ZO gradient at reference point $\hat{g}_t^{\text{ref}}$,

Algorithm Structure: ZO-AdaMM-SVRG ii

  - variance-reduced gradient $g_{s,t}^{\mathrm{VR}} = \hat{g}_t - \hat{g}_t^{\mathrm{ref}} + \tilde{g}_s$.
- Update momentum and second moments (AdaMM + AMSGrad).
- Apply adaptive parameter update.

After m inner steps:

$$x_{s+1} = x_{s,m}.$$

ZO-AdaMM-SVRG Algorithm

Algorithm 3 ZO-AdaMM-SVRG

Require: Initial point x_0, smoothing parameter μ, inner loop length m, number of random directions q, learning rate schedule $\{\alpha_t\}$, Adam parameters β_1, β_2, small $\varepsilon > 0$.

Initialize $m_0 = 0$, $v_0 = v_{\text{init}}\mathbf{1}$, $\hat{v}_0 = v_0$.

for $s = 0, 1, 2, \ldots, S-1$ **do**

$x_{s,0} = x_s$

Compute a **reference gradient** (large batch):

$$\bar{g}_s = \frac{d}{Q}\sum_{i=1}^{Q} \frac{f(x_{s,0} + \mu u_i) - f(x_{s,0} - \mu u_i)}{2\mu}\, u_i,$$

where $u_i \sim \mathcal{N}(0, I_d)$ normalized.

Set $m_{s,0} = 0$, $v_{s,0} = v_{\text{init}}\mathbf{1}$, $\hat{v}_{s,0} = v_{s,0}$.

for $t = 0, 1, \ldots, m-1$ **do**

Sample q random directions $u_{t,1}, \ldots, u_{t,q}$.

Compute **ZO gradient at current point**:

$$\hat{g}_t = \frac{d}{q}\sum_{i=1}^{q} \frac{f(x_{s,t} + \mu u_{t,i}) - f(x_{s,t} - \mu u_{t,i})}{2\mu}\, u_{t,i}.$$

Compute **ZO gradient at reference point**:

$$\hat{g}_t^{\text{ref}} = \frac{d}{q}\sum_{i=1}^{q} \frac{f(x_{s,0} + \mu u_{t,i}) - f(x_{s,0} - \mu u_{t,i})}{2\mu}\, u_{t,i}.$$

Variance–Reduced ZO gradient:

$$g_{s,t}^{\text{VR}} = \hat{g}_t - \hat{g}_t^{\text{ref}} + \bar{g}_s.$$

Adam first moment:

$$m_{s,t+1} = \beta_1 m_{s,t} + (1-\beta_1) g_{s,t}^{\text{VR}}.$$

Adam second moment:

$$v_{s,t+1} = \beta_2 v_{s,t} + (1-\beta_2)\big(g_{s,t}^{\text{VR}}\big)^2.$$

AMSGrad correction:

$$\hat{v}_{s,t+1} = \max(\hat{v}_{s,t}, v_{s,t+1}).$$

Adaptive update:

$$x_{s,t+1} = x_{s,t} - \alpha_{s,t}\frac{m_{s,t+1}}{\sqrt{\hat{v}_{s,t+1}} + \varepsilon}.$$

Projection (if constrained):

$$x_{s,t+1} \leftarrow \Pi_{\mathcal{X}}(x_{s,t+1}).$$

end for

Set $x_{s+1} = x_{s,m}$.

end for

return Final iterate x_S.

Evaluation: Margin-Based Black-Box Attack

We evaluate our optimizer using a standard **black-box adversarial attack** where no gradients are available. The model is accessed only through `Predict(·)` queries.

Given an input x_0 with true label y, the goal is to find a bounded perturbation δ such that:

$$\tilde{x} = x_0 + \delta$$

reduces the classifier's confidence in the true class relative to the most likely incorrect class.

The attack minimizes the objective:

$$f(\delta) = M(\delta) + \lambda D(\delta)$$

where:

- $M(\delta)$: margin encouraging misclassification,
- $D(\delta) = \|\delta\|_2$: distortion penalty.

Evaluation: Margin-Based Attack

Margin term:

$$M(\delta) = \log(p_{\text{true}} + 10^{-12}) - \log(p_{\text{other}} + 10^{-12})$$

where:

- p_{true}: predicted probability of the true class,
- p_{other}: highest probability among incorrect classes.

A projection operator $\Pi_{[l,u]}$ keeps perturbations within valid pixel bounds.

Early stopping improves query efficiency:

- margin threshold reached,
- distortion limit exceeded,
- no improvement for a patience window,
- moving-average plateau,
- query budget exhausted.

Experimental Setup: Experiment 1

Black-Box Attack Evaluation: All ZO optimizers update the perturbation $\delta = x - x_0$ using their respective gradient estimates, then apply: $\delta_{t+1} = \text{Proj}_{\|\delta\| \leq \epsilon}(\delta_t - \alpha\, \hat{g}(x))$.

The adversarial input is reconstructed as: $x_{t+1} = x_0 + \delta_{t+1}$.

Metrics: Query count, confidence drop, perturbation norm, and attack success rate.

Dataset: Wisconsin Breast Cancer dataset (569 samples, 30 continuous diagnostic features such as radius, texture, perimeter, smoothness, etc.).

Model: A fully connected neural network trained as the target classifier:

- Input: 30-dimensional feature vector,
- Hidden layers: 64 and 32 units (ReLU),
- Dropout for regularization,
- Sigmoid output for binary classification (benign / malignant).

This model serves as the **black-box classifier** for the attack.

Experimental Setup: Experiment 2

Datasets:

- MNIST (10 classes, grayscale images)
- CIFAR-10 (10 classes, RGB images)

Models:

- **MNIST:** Standard CNN with:
  - 2 convolutional layers (32 and 64 filters, 3×3 kernels),
  - ReLU activations + 2×2 max-pooling,
  - Fully connected layer with 128 units,
  - 10-class output.

  Achieves **99%** test accuracy.
- **CIFAR-10:** Modified ResNet-18:
  - 3×3 first convolution,
  - No max-pool layer,
  - 10-class fully connected head.

  Achieves **93–95%** test accuracy.

Experimental Setup: Experiment 3

Dataset: CIFAR-100 (100 classes, RGB images). Chosen because its higher class diversity creates a more challenging black-box attack setting than CIFAR-10.

Model: Pretrained **WideResNet-28** (WRN-28), a widened variant of ResNet that increases channel width for higher representation capacity.

- Depth: 28 layers
- Fine-tuned on CIFAR-100 using standard preprocessing
- Training includes:
  - SGD with momentum 0.9
  - Learning rate 0.1, weight decay 5×10^{-4}
  - Standard CIFAR augmentations (random crop, flip)

The fine-tuned model reaches accuracy comparable to SOTA WRN-28-10.

Metrics: Success rate, average queries, confidence drop, perturbation norm.

Results: Experiment 1 — Model Performance

Neural Network Evaluation

Metric	Value
Accuracy	0.9708
Precision	0.9811
Recall	0.9720
F1-Score	0.9765

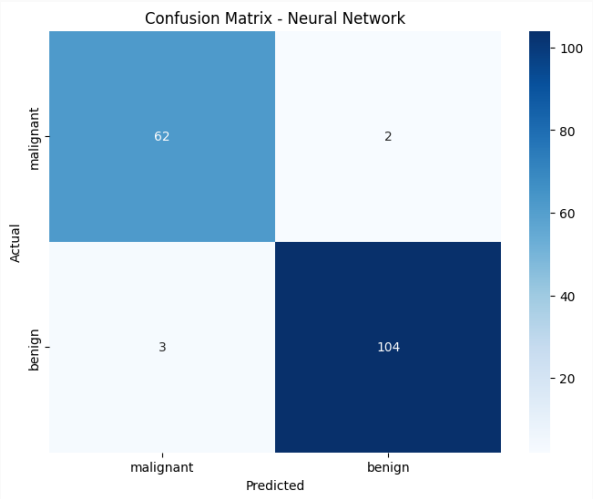

Figure 1: Confusion matrix of the trained model.

Results: Experiment 1 — Attack Performance

Black-box attack results:

Method	Success	Avg Queries	Avg Conf Drop
ZO-AdaMM	75	114	-0.3497
ZO-SGD	79	2796	-0.4046
VR-ZO-AdaMM	78	510	-0.3681

Results: Experiment 1 — Query Perturbation Plots

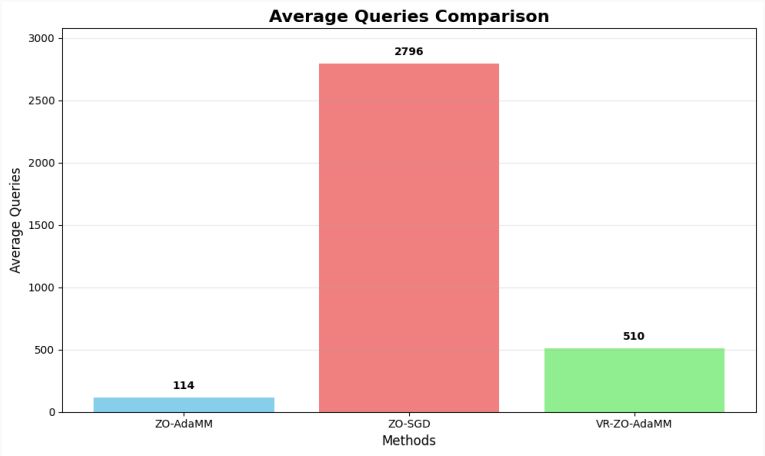

Figure 2: Average query count across attack methods.

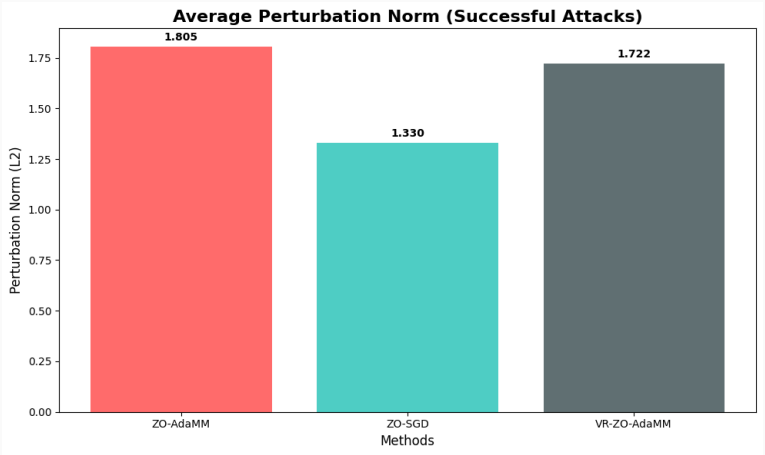

Figure 3: Average perturbation magnitude across methods.

Results: Experiment 2 — Attack Performance

Model Accuracy:

- MNIST CNN: 99% accuracy
- CIFAR-10 ResNet-18: 93–95% accuracy

Method	Success	Perturbation	Avg Queries
ZO-AdaMM SVRG (MNIST)	0.999	6.5	1177
ZO-AdaMM (MNIST)	0.999	7.2	220
ZO-AdaMM SVRG (CIFAR-10)	0.967	32.0	1524
ZO-AdaMM (CIFAR-10)	0.705	28.9	587

Results: Experiment 3 — CIFAR-100 (WRN-28)

Fine-tuning:

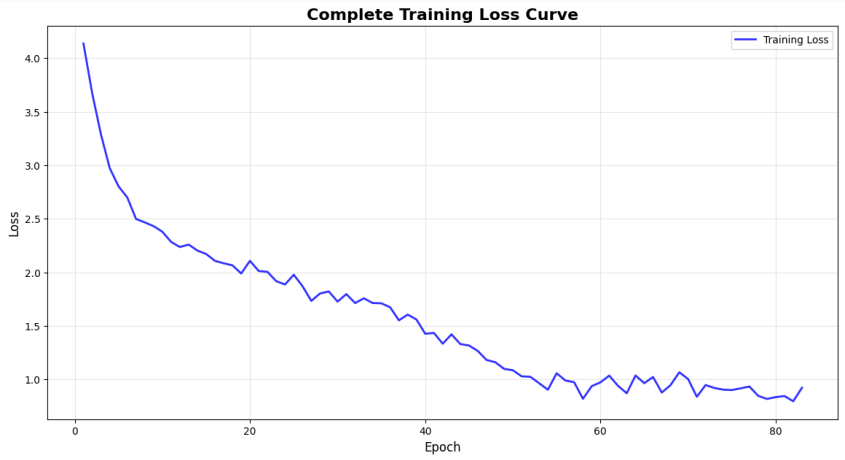

Figure 4: Loss curve during model fine-tuning.

Black-box attack results:

Method	Success Rate	Avg Queries	Avg Conf Drop
ZO-AdaMM	100.0%	106	0.2768
ZO-SGD	60.0%	1710	0.1984
VR-ZO-AdaMM	100.0%	201	0.3612

Results: Experiment 3 — Query Perturbation Plots

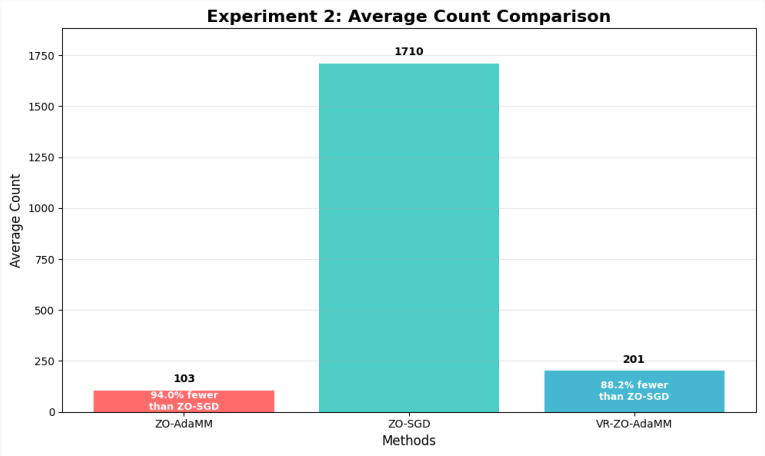

Figure 5: Average number of queries.

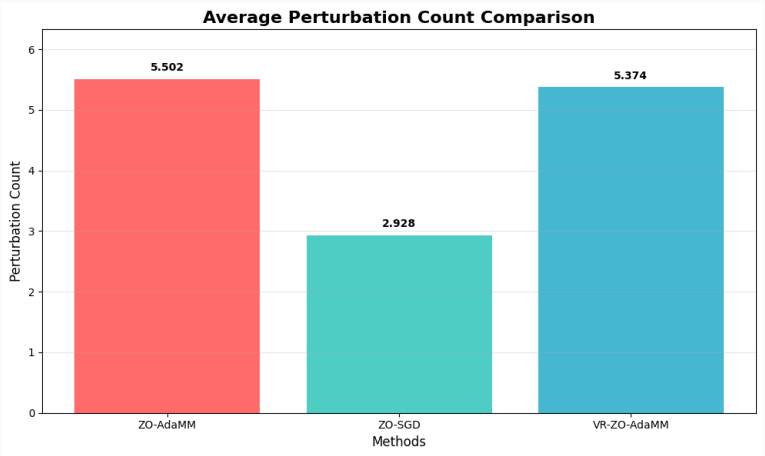

Figure 6: Average perturbation magnitude.

Conclusion

Across all experiments, clear patterns emerge:

- **ZO-SGD** requires very high query counts due to noisy gradient estimates.
- **ZO-AdaMM** improves stability through momentum and adaptive scaling, reducing queries substantially.
- **VR-ZO-AdaMM** further lowers variance in gradient estimates, achieving:
  - stronger confidence drop,
  - competitive success rates,
  - low query requirements.

Overall: VR-ZO-AdaMM provides the best balance between attack strength and query efficiency, making it the most effective method in practical black-box adversarial settings.