MVT-ATTACK: MOMENTUM VARIANCE TUNING FOR TRANSFERABLE POINT CLOUD ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

The vulnerability of 3D deep learning models has garnered significant attention. However, existing point cloud adversarial attacks exhibit limited transferability due to insufficient modeling of gradient dynamics. Instantaneous gradients provide limited characterization of local optimization uncertainty, while cross-iteration gradient evolution remains insufficiently exploited, leading to unstable optimization trajectories. To address this issue, we propose MVT-Attack (Momentum Variance Tuning Attack), a novel framework that enhances attack transferability through a dual gradient regulation mechanism. Unlike previous approaches that rely on complex loss-function regularization, MVT-Attack directly regulates gradient dynamics through a two-stage optimization strategy. First, we introduce a variance-guided gradient rectification mechanism that estimates local gradient fluctuations via normal-constrained sampling and adaptively adjusts gradients to reduce source-model-specific optimization bias. Subsequently, we design a momentum optimization module that aggregates historically refined gradients to construct smoother and more consistent update directions. The proposed framework demonstrates the effectiveness of gradient dynamics regulation to improve attack transferability and achieves superior performance on the ModelNet40 and ScanObjectNN benchmarks, consistently outperforming state-of-the-art methods. The code will be made publicly available upon acceptance.

1 INTRODUCTION

Point clouds, as a fundamental 3D representation modality, serve as a critical medium for intelligent systems to perceive the physical world, with broad applications in autonomous driving (Zhou & Tuzel, 2018; Qi et al., 2018), robotic navigation (Liang et al., 2019; De Sa et al., 2024), and semantic segmentation (Li et al., 2025a; 2020). While deep neural networks have achieved remarkable performance in point cloud analysis, their robustness limitations have raised increasing concerns. Adversarial attacks (Zheng et al., 2019; Wicker & Kwiatkowska, 2019), which introduce imperceptible perturbations to manipulate model predictions, expose intrinsic vulnerabilities in geometric perception and provide valuable insights into model robustness and decision boundaries. Consequently, generating transferable adversarial examples capable of attacking unseen models has become an important problem for evaluating the robustness of 3D perception systems.

Existing point cloud adversarial attack methods have primarily focused on two complementary objectives: imperceptibility and transferability. Regarding imperceptibility, researchers have explored two major paradigms for constraining perturbations. First, geometric-aware regularization terms are incorporated into loss functions to preserve local structural integrity, as demonstrated by GeoA^3 (Wen et al., 2020) with geometric priors, KNN (Tsai et al., 2020) with K-nearest neighbor topology, MGA (Liu & Hu, 2024) with multi-granularity spectral filtering, and manifold-constrained approaches (Tang et al., 2024). Second, explicit geometric constraints are imposed at the gradient level; for example, SI-Adv (Huang et al., 2022) restricts perturbations to local tangent planes to preserve shape semantics, while CIM (Zhang et al., 2024a) adaptively calibrates gradients according to curvature anisotropy.

For transferability, existing methods mainly improve generalization through sophisticated loss-function regularization. Representative approaches include PF-Attack (He et al., 2023), which extracts effective information from decomposed sub-perturbations, and CFG (Pang et al., 2025), which

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

guides attack directions using cross-model critical features. However, these approaches largely rely on instantaneous gradient optimization, lacking effective modeling of local gradient fluctuations and cross-iteration gradient evolution. Consequently, the optimization trajectory remains vulnerable to source-model-specific bias and suboptimal convergence. Moreover, gradient-level optimization mechanisms specifically designed to enhance transferability in 3D point cloud adversarial attacks remain relatively underexplored, motivating our proposed framework.

To overcome the limitations of existing methods, we propose MVT-Attack, which avoids additional complex loss-function regularization and introduces a dynamic regulation mechanism at the gradient optimization level. By jointly exploiting local gradient variance and historical gradient information, MVT-Attack alleviates source-model-specific optimization bias, stabilizes the optimization trajectory, and guides adversarial perturbations toward regions with improved cross-model transferability. As illustrated in Fig. 1.

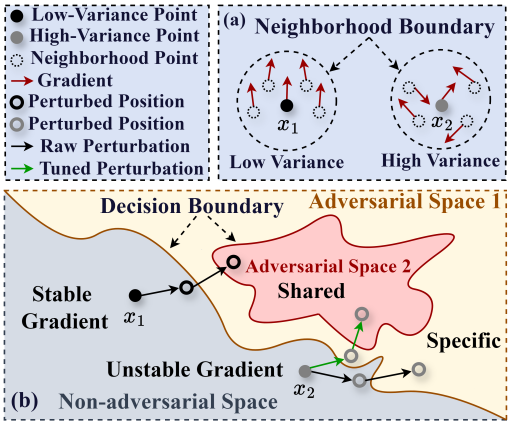

Figure 1: **(a) Gradient variance analysis.** Points with low gradient variance (e.g., x_1) exhibit consistent gradients, indicating stable optimization directions, whereas points with high gradient variance (e.g., x_2) show dispersed gradients, reflecting stronger source-model-specific bias. **(b) Gradient rectification.** Near the decision boundary, raw update directions (black arrows) at low-variance x_1 naturally approach the *Shared Adversarial Space*, while those at high-variance x_2 tend to drift toward the *Model-Specific Space*. MVT-Attack exploits gradient variance and momentum to rectify optimization trajectories, steering perturbations toward the shared space for improved transferability. The rectified trajectories are shown by green arrows in the color version.

Specifically, we first design a variance modulation mechanism that constructs consistent perturbed point clouds through geometry-aware sampling with normal constraints. This enables reliable estimation of local gradient variance to characterize optimization uncertainty. Then, we introduce an adaptive gradient correction term that incorporates variance information into the original gradient, adaptively adjusting unstable gradient components and guiding the optimization trajectory toward flatter regions. Building upon this, we propose a momentum optimization module that applies normalized exponential weighting to variance-corrected gradient sequences. This generates a historically consistent and noise-robust aggregated gradient, reducing the accumulation of unstable optimization information in conventional momentum schemes. Ultimately, the update direction (green arrow in Fig. 1(b)) exhibits improved stability across model-specific decision boundaries and moves toward transferable adversarial regions shared across models, achieving high-transferability attacks. Our contributions are threefold:

- We propose MVT-Attack, a gradient-level optimization framework that jointly exploits local gradient variance and historical gradient information to stabilize optimization trajectories and improve attack transferability.
- We propose a normal-constrained sampling strategy tailored to point clouds, which leverages surface normal constraints to generate geometrically consistent perturbed point clouds for reliable gradient variance estimation.
- Extensive experiments on ModelNet40 and ScanObjectNN demonstrate that MVT-Attack consistently outperforms existing methods in black-box attack scenarios.

2 Related Work

2.1 Imperceptibility-focused Attacks.

3D-Adv (Xiang et al., 2019) was among the first to systematically investigate point cloud adversarial attacks by introducing point perturbation and point generation paradigms. However, the generated adversarial examples often contain noticeable outliers, making them vulnerable to outlier removal

defenses. To alleviate this issue, GeoA^3 (Wen et al., 2020) incorporated geometry-aware constraints with local curvature consistency to preserve surface smoothness, improving imperceptibility and robustness against geometric defenses. Subsequently, GSDA (Hu et al., 2022) explored spectral-domain perturbations by exploiting low-frequency geometric components for more natural adversarial generation. To further enhance perturbation concealment, HiT-Adv (Lou et al., 2024) hides perturbations in high-curvature regions, while SCP (Tang et al., 2026) exploits synergistic effects among sparse points to achieve effective attacks with reduced perturbation magnitude. However, these methods primarily focus on geometric imperceptibility under white-box settings, and their transferability to unseen models remains limited.

2.2 Transferability-focused Attacks.

AdvPC (Hamdi et al., 2020) explored autoencoder-based reconstruction with data distribution constraints to improve cross-model transferability. To decouple transferable components from universal perturbations, PF-Attack (He et al., 2023) introduced stochastic perturbation factorization to identify transferable perturbation components. SS-Attack (Zhang et al., 2024b) demonstrated that scale and shear transformations can improve attack robustness across different geometric transformations. Furthermore, 3DAdvDiff (Dai & Xiao, 2024) embedded adversarial perturbations into diffusion-based shape completion by leveraging generative shape priors, achieving strong transferability. Recently, NoPain (Li et al., 2025b) proposed a no-box attack paradigm that generates adversarial examples under target-model-agnostic settings through optimal transport singular boundaries, while CFG (Pang et al., 2025) achieved high-success black-box transfer by perturbing cross-model shared geometric features. However, existing methods still lack explicit modeling of gradient dynamics, where instantaneous gradients fail to characterize local optimization uncertainty and cross-iteration gradient evolution, limiting further improvements in transferability.

2.3 Adversarial Gradient Optimization

For 2D adversarial attacks, gradient optimization has been widely explored to improve transferability. MI-FGSM (Dong et al., 2018) introduces momentum to stabilize update directions, while VMI-FGSM (Wang & He, 2021) incorporates gradient variance estimation to reduce update uncertainty and enhance transferability. In 3D point cloud attacks, gradient-based optimization has mainly focused on preserving geometric structures. SI-Adv (Huang et al., 2022) constrains gradients within local tangent planes to preserve shape semantics, while CIM (Zhang et al., 2024a) adaptively adjusts gradients according to curvature anisotropy for geometric-aware perturbation generation. However, these approaches primarily target imperceptibility rather than transferability. Gradient optimization strategies explicitly designed for improving transferability in 3D point cloud attacks remain relatively underexplored, motivating our work.

3 Methodology

This section presents MVT-Attack, an efficient transferability optimization framework for point cloud adversarial attacks. Instead of improving transferability through explicit loss regularization, MVT-Attack enhances gradient optimization via geometry-aware variance tuning and momentum-guided gradient rectification. The framework is illustrated in Fig. 2.

3.1 Preliminaries

Formalization of Adversarial Attack. Given a benign point cloud sample $X \in \mathbb{R}^{N\times 3}$ and its true label y, the goal is to find a minimal perturbation Δ such that the perturbed sample $X^{\text{adv}} = X + \Delta$ is misclassified by model $f(\cdot)$. This is typically formulated as:

$$\min_{X^{\text{adv}}} \; -\mathcal{L}_{\text{cls}}(f(X^{\text{adv}}), y), \quad \text{s.t.} \quad D(X, X^{\text{adv}}) \le \epsilon, \tag{1}$$

where $\mathcal{L}_{\text{cls}}$ is a classification loss (e.g., cross-entropy) guiding optimization toward misclassification, and $D(\cdot)$ measures the deviation between X and X^{adv}, commonly instantiated as an ℓ_p norm (ℓ_2 or ℓ_∞) to constrain perturbation magnitude and ensure imperceptibility.

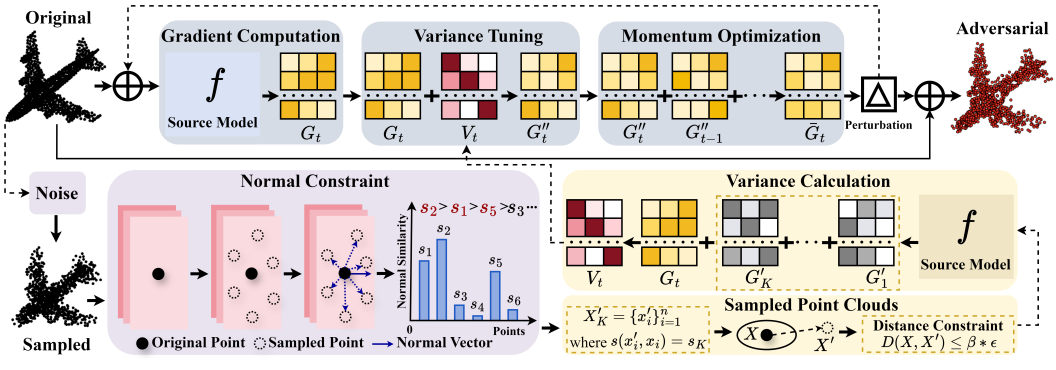

Figure 2: MVT-Attack framework comprising two core modules. (1) Variance tuning employs normal-constrained geometry-aware sampling to prevent geometric noise from contaminating gradient estimation, ensuring accurate variance computation. (2) Momentum optimization accumulates historical gradients to construct smooth and stable update trajectories.

Transfer-based Attacks. Many methods (He et al., 2023; Pang et al., 2025; Chen et al., 2026) incorporate a transfer-aware regularizer $\mathcal{R}$ to alleviate source-model overfitting:

$$\min_{X^{\text{adv}}} -\mathcal{L}_{\text{cls}}(f(X^{\text{adv}}), y) + \gamma \cdot \mathcal{R}(X, X^{\text{adv}}), \quad \text{s.t.} \quad D(X, X^{\text{adv}}) \leq \epsilon, \tag{2}$$

where $\mathcal{R}$ encodes transferability priors (e.g., key-feature guidance in CFG (Pang et al., 2025) leveraging shared structures between source and target models), and γ balances attack efficacy and regularization.

Discussion. Although existing methods alleviate source-model overfitting to some extent, they still rely on instantaneous gradients, making the optimization process sensitive to local noise and prone to converging to sharp minima with poor transferability. Instead of introducing an explicit regularization term $\mathcal{R}$, MVT-Attack rectifies the optimization direction by exploiting local gradient variability. As detailed in the next subsection, this variability is quantified through geometry-aware neighborhood sampling, enabling smoother optimization trajectories and improved transferability.

3.2 Variance Tuning

To address the limitations of instantaneous gradients, we propose a variance-aware gradient rectification mechanism. Specifically, we quantify local gradient uncertainty through structure-aware neighborhood sampling and leverage the resulting gradient variance to adaptively rectify the optimization direction, guiding adversarial examples toward flatter regions with improved transferability. We first formalize the notion of gradient variance.

Definition 1 ***Gradient Variance.*** *Given a classifier f and the untargeted attack loss $\mathcal{L}(f(X), y) \triangleq -\mathcal{L}_{\text{cls}}(f(X), y)$, let $X \in \mathbb{R}^{N\times 3}$ be an input point cloud. The gradient variance $V(X)$ is defined as the deviation between the gradient at X and the expected gradient within a local perturbation neighborhood bounded by $\epsilon' = \beta \cdot \epsilon$:*

$$V(X) = \mathbb{E}_{\|X'-X\|_p<\epsilon'}[\nabla_{X'}\mathcal{L}(f(X'), y)] - \nabla_X \mathcal{L}(f(X), y), \tag{3}$$

where β is the sampling control coefficient and ϵ denotes the imperceptibility threshold.

Since the expectation in Eq. 3 cannot be computed analytically, we approximate it via Monte Carlo sampling using K neighboring point clouds:

$$V(X) \approx \frac{1}{K}\sum_{j=1}^{K} \nabla_{X'_j}\mathcal{L}(f(X'_j), y) - \nabla_X \mathcal{L}(f(X), y). \tag{4}$$

The approximation quality depends on the sampled neighboring point clouds $\{X'_j\}_{j=1}^{K}$. Conventional random sampling often introduces outliers detached from the object surface, contaminating gradient estimation with geometric noise. To better preserve the geometric structure of point clouds, we next introduce a geometry-aware sampling strategy.

Normal-Constrained Sampling. To align the sampling process with the subsequent perturbation optimization, we constrain perturbation sampling using the local surface normal. Specifically, for each point x_i, multiple candidate points are generated by adding random perturbations. For each candidate point x'_j, we compute its displacement vector with respect to the original point:

$$\boldsymbol{d}_j = x'_j - x_i. \tag{5}$$

Since the optimized perturbation is ultimately projected onto the local surface normal, we evaluate each candidate according to the cosine similarity between its displacement direction and the local surface normal:

$$s_j = \frac{\boldsymbol{n}_{x_i}^\top \boldsymbol{d}_j}{\|\boldsymbol{n}_{x_i}\|_2 \|\boldsymbol{d}_j\|_2}, \tag{6}$$

where a larger s_j indicates better alignment with the local surface normal. The candidate points of each point are ranked in descending order of s_j, and the k-th ranked candidates are aggregated to form the k-th neighboring point cloud X'_k, yielding K neighboring point clouds $\{X'_1, \ldots, X'_K\}$. This strategy maintains geometric consistency while preserving perturbation diversity, enabling more reliable estimation of gradient variance.

Variance-Guided Gradient Rectification. At each optimization iteration, based on the sampled neighboring point clouds, we first compute the gradients $\{G'_1, \ldots, G'_K\}$ and approximate the local gradient variance according to Eq. 3:

$$V(X) = \frac{1}{K} \sum_{j=1}^{K} G'_j - G, \tag{7}$$

where $G = \nabla_X \mathcal{L}(f(X), y)$ and $G'_j = \nabla_{X'_j} \mathcal{L}(f(X'_j), y)$.

The estimated gradient variance measures the discrepancy between the current gradient and the expected local gradient, thereby characterizing the local uncertainty of the optimization direction. Higher gradient variance indicates that the current update direction is less stable and more susceptible to overfitting to the source model.

To exploit this information, we construct a variance-aware adaptive correction term and rectify the current gradient as

$$G'' = G + \lambda \cdot \frac{V(X)}{\|V(X)\|_1 + p}, \tag{8}$$

where λ controls the correction strength and p is a small constant for numerical stability. The corrected gradient incorporates local gradient information from neighboring point clouds, yielding a smoother optimization trajectory and improved attack transferability.

3.3 Momentum Optimization

Although variance-guided gradient rectification improves the robustness of individual updates, optimization in high-dimensional point cloud spaces may still suffer from stochastic fluctuations and irregular loss landscapes. To further enhance cross-iteration consistency, we incorporate a momentum mechanism that aggregates historical variance-corrected gradients for a smoother optimization trajectory.

Specifically, the aggregated gradient at iteration t is defined as the normalized exponentially weighted accumulation of the available variance-corrected gradients.

$$\bar{G}_t = \frac{\sum_{i=0}^{min(L-1,t-1)} \mu^{i+1} G''_{t-i}}{\sum_{i=0}^{min(L-1,t-1)} \mu^{i+1}}, \tag{9}$$

where μ denotes the momentum coefficient and L represents the historical window length. The normalization reduces the influence of μ and L on the overall gradient magnitude, allowing the momentum mechanism to focus on adjusting the relative contribution of historical gradients and stabilizing the optimization direction. Finally, the perturbation is updated as:

$$\Delta_t = \Delta_{t-1} - \eta \cdot \bar{G}_t, \tag{10}$$

where η denotes the step size.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Implementation. Following the protocol of CFG (Pang et al., 2025), we set the maximum iteration number for the update of the point cloud perturbation at $T = 200$ with a step size of $\eta = 0.01$. For variance regulation, we configure the balancing hyperparameter $\lambda = 1.7$, sampling threshold $\beta = 1.7$, and the number of sampled point clouds $K = 7$. For momentum regulation, the gradient history window length is set to $L = 10$ with an influence coefficient $\mu = 0.9$. All experiments are conducted on an RTX 4090D GPU.

Datasets. We evaluate the performance and transferability of the proposed method on two benchmark datasets: ModelNet40 and ScanObjectNN. ModelNet40 (Wu et al., 2015) is a widely used synthetic CAD dataset comprising 12,311 samples in 40 categories. ScanObjectNN (Uy et al., 2019) provides more challenging real-world scanned point clouds, covering 15 categories with complex background noise and occlusions. All samples are uniformly downsampled to 1,024 points and normalized to a unit cube.

Victim Models. To evaluate cross-architecture transferability, we employ PointNet (Qi et al., 2017a), PointNet++ (SSG/MSG) (Qi et al., 2017b), and DGCNN (Wang et al., 2019) as source models for perturbation generation. Beyond these source models, the victim models encompass mainstream 3D paradigms, including convolution/graph networks (PointConv (Wu et al., 2019), PointCNN (Li et al., 2018)), Transformers (PCT (Guo et al., 2021), PT (Zhao et al., 2021)), curve networks (CurveNet (Xiang et al., 2021)), and non-parametric models (Point-PN (Zhang et al., 2023)). This heterogeneous setup rigorously validates the generalizability of our attack across diverse feature extraction mechanisms.

Baselines. We compare our method against four representative baselines. Although 3D-Adv (Xiang et al., 2019) and GeoA^3 (Wen et al., 2020) are white-box attacks, we evaluate them under a transfer-based setting. PF-Attack (He et al., 2023) and CFG (Pang et al., 2025) are classic transfer-based black-box attacks, which generate perturbations via source model gradients without accessing the target model. This setup comprehensively covers mainstream attack paradigms from white-box to black-box.

Evaluation Metrics. Following the experimental protocols of CFG (Pang et al., 2025), we employ Attack Success Rate (ASR) and Transferability Score (TS) to evaluate adversarial performance and transferability. Specifically, ASR measures the effectiveness of an attack on a specific model, while TS reflects the overall transferability by averaging the ASRs across all target models. Additionally, we use Average Time (A.T) to measure the computational efficiency of generating adversarial examples.

4.2 PERFORMANCE EVALUATION

Transferability. We systematically evaluate the cross-architecture transferability of MVT-Attack on 3D point cloud models. As shown in Tab. 1 and Tab. 2, when using PointNet and PointNet++(MSG) as source models, MVT-Attack achieves the highest attack success rates across all six target models, significantly outperforming all baseline methods. Tab. 3 further reports the transferability scores (TS): on ModelNet40, MVT-Attack attains 49.0% ($\epsilon = 0.18$) and 68.4% ($\epsilon = 0.45$), surpassing the state-of-the-art CFG (Pang et al., 2025) by 7.6% and 13.0%, respectively; on the more challenging ScanObjectNN dataset, it reaches 60.9% and 67.2%, with improvements of 9.4% and 13.1%. These results demonstrate that the collaborative design of neighborhood variance modulation and momentum optimization effectively mitigates source model overfitting, thereby substantially enhancing the transferability of adversarial examples. Results with additional source models and target models are provided in the appendix.

Robustness. As shown in Tab. 4, we evaluate the robustness of MVT-Attack with PointNet as the source model against four defense mechanisms (SRS (Yang et al., 2019), SOR (Zhou et al., 2019), DUP-Net (Zhou et al., 2019), and CCN (Li et al., 2022)). On ModelNet40, our method achieves the best performance under SOR and CCN defenses. Under SRS and DUP-Net, it is slightly inferior to CFG but still significantly outperforms other baselines. On ScanObjectNN, due to real-world scanning noise and small inter-class geometric variations, the feature extraction capability of defense

Table 1: Attack transferability across classical models on ModelNet40. Best results are in **bold**.

ϵ	Source model	Attack method	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	Point CNN	PCT	Point -PN
0.18	PointNet	3D-Adv	100.0	8.4	10.4	6.8	1.0	0.0	1.2
		GeoA^3	100.0	20.0	19.6	7.2	2.1	3.3	5.5
		PF-Attack	100.0	49.6	61.5	24.8	29.5	23.4	26.1
		CFG	100.0	72.6	80.6	39.3	32.7	30.4	42.5
		MVT-Attack	100.0	**73.5**	**84.0**	**46.8**	**46.0**	**41.4**	**48.5**
	PointNet++ (MSG)	3D-Adv	6.8	100.0	28.4	11.2	1.0	1.8	5.7
		GeoA^3	4.4	100.0	14.4	6.4	4.4	0.0	6.9
		PF-Attack	17.2	100.0	67.0	24.3	22.2	16.6	28.9
		CFG	17.6	100.0	72.9	25.3	21.0	26.4	35.1
		MVT-Attack	**26.1**	100.0	**75.0**	**33.1**	**36.6**	**33.8**	**43.0**
0.45	PointNet	3D-Adv	100.0	8.8	9.6	8.0	0.1	0.0	0.7
		GeoA^3	100.0	23.6	20.8	7.2	5.5	7.5	10.2
		PF-Attack	100.0	59.3	64.8	28.8	33.4	28.7	40.8
		CFG	100.0	84.7	88.0	49.6	43.4	48.6	61.5
		MVT-Attack	100.0	**89.7**	**93.8**	**67.6**	**68.7**	**64.9**	**76.2**
	PointNet++ (MSG)	3D-Adv	7.2	100.0	29.2	11.2	0.5	1.8	8.7
		GeoA^3	4.4	100.0	13.6	6.0	4.7	0.0	4.2
		PF-Attack	19.2	100.0	75.0	27.5	25.5	16.0	35.3
		CFG	36.6	100.0	83.5	34.7	26.2	47.5	61.3
		MVT-Attack	**52.0**	100.0	**86.2**	**49.0**	**52.0**	**56.9**	**63.4**

Table 2: Attack transferability across classical models on ScanObjectNN. Best results are in **bold**.

ϵ	Source model	Attack method	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	Point CNN	PCT	Point -PN
0.18	PointNet	3D-Adv	100.0	17.1	15.7	6.8	11.4	8.2	16.8
		GeoA^3	100.0	23.4	24.8	18.1	19.0	19.7	30.7
		PF-Attack	100.0	59.7	57.4	26.5	39.5	29.1	48.7
		CFG	100.0	60.5	61.5	30.1	40.7	35.9	52.3
		MVT-Attack	100.0	**66.9**	**61.5**	**51.3**	**60.4**	**52.0**	**61.8**
	PointNet++ (MSG)	3D-Adv	14.0	100.0	13.7	13.8	13.6	7.3	19.1
		GeoA^3	17.0	100.0	16.2	14.0	16.8	10.7	15.8
		PF-Attack	47.2	100.0	57.7	33.7	42.7	30.3	47.2
		CFG	52.6	100.0	74.8	44.9	53.5	50.7	60.1
		MVT-Attack	**62.2**	100.0	**78.2**	**54.6**	**63.7**	**57.4**	**61.3**
0.45	PointNet	3D-Adv	100.0	17.2	14.3	7.5	12.2	8.1	18.1
		GeoA^3	100.0	20.8	19.9	15.8	18.3	17.2	26.9
		PF-Attack	100.0	59.0	59.0	28.1	40.1	32.1	54.2
		CFG	100.0	60.3	61.7	32.9	38.9	35.9	59.1
		MVT-Attack	100.0	**71.3**	**75.4**	**55.9**	**61.9**	**54.0**	**65.9**
	PointNet++ (MSG)	3D-Adv	14.2	100.0	14.1	13.4	13.4	7.5	19.9
		GeoA^3	17.0	100.0	16.0	14.0	15.4	10.1	16.4
		PF-Attack	48.0	100.0	57.9	33.1	43.7	30.3	46.3
		CFG	60.0	100.0	77.2	47.5	51.1	54.9	69.2
		MVT-Attack	**71.6**	100.0	**82.4**	**62.1**	**68.5**	**63.6**	**73.3**

Table 3: Transferability Score on ModelNet40 and ScanObjectNN.

Dataset	ϵ	Transferability Score (%)				
		3D-Adv	GeoA^3	PF-Attack	CFG	**MVT-Attack**
ModelNet40	0.18	6.9	7.9	32.6	41.4	**49.0**
	0.45	7.2	9.0	37.9	55.4	**68.4**
ScanObjectNN	0.18	13.1	18.9	43.3	51.5	**60.9**
	0.45	13.3	17.3	44.3	54.1	**67.2**

Table 4: Robustness evaluation against different defense strategies on ModelNet40 and ScanObjectNN. Results are attack success rates (%).

Method	ModelNet40								ScanObjectNN							
	$\epsilon = 0.18$				$\epsilon = 0.45$				$\epsilon = 0.18$				$\epsilon = 0.45$			
	SRS	SOR	DUP-Net	CCN	SRS	SOR	DUP-Net	CCN	SRS	SOR	DUP-Net	CCN	SRS	SOR	DUP-Net	CCN
3D-Adv	26.4	17.2	12.4	0.0	1.0	17.2	12.9	0.0	98.1	97.4	97.9	10.3	98.3	97.1	96.4	8.7
GeoA^3	58.3	43.8	19.7	11.3	53.8	40.8	23.5	11.7	**98.9**	**98.4**	98.2	20.9	**98.7**	98.2	97.3	15.6
PF-Attack	55.7	42.4	25.2	30.0	61.1	33.8	31.1	36.5	98.5	98.2	**98.5**	34.7	98.3	**98.8**	**98.3**	36.1
CFG	**79.2**	62.9	**61.6**	50.6	**87.9**	73.0	**73.2**	66.8	98.6	98.1	98.1	39.8	98.6	97.2	97.2	45.9
MVT-Attack	74.1	**67.3**	58.1	**57.9**	83.0	**75.1**	64.1	**78.9**	98.1	**98.4**	96.7	**53.7**	98.2	98.4	97.4	**59.1**

Table 5: Module contributions with PointNet as the source model.

VT	Mom	NC	PointNet++ (SSG)	PointNet++ (MSG)	DGCNN	PointCNN	PCT
			19.3	10.0	4.2	14.6	4.6
✓			90.0	86.1	61.5	62.6	58.6
	✓		84.1	79.3	43.9	45.3	45.1
✓	✓		91.5	88.9	62.2	65.5	59.4
✓		✓	**93.8**	89.5	64.7	63.6	62.3
✓	✓	✓	**93.8**	**89.7**	**67.6**	**68.7**	**64.9**

models is limited, resulting in high attack success rates for all methods and narrowed performance gaps. Nevertheless, MVT-Attack maintains performance comparable to or better than the optimal baselines in this complex scenario, demonstrating its robustness and generalization across different defenses and data distributions.

Visualization. We further visualize the adversarial samples generated by MVT-Attack. As shown in Fig. 3, we compare the clean and adversarial point clouds across four source models (PointNet, PointNet++(MSG/SSG), and DGCNN) on ModelNet40 and ScanObjectNN. We observe that the adversarial samples maintain high geometric consistency with the original ones, with perturbations primarily concentrated on edges or sparse regions, manifesting as subtle local deformations rather than global noise. Despite the minimal visual discrepancy, all target models are successfully misled. This intuitively confirms that MVT-Attack can effectively deceive diverse 3D perception models via minor geometric distortions while preserving semantic integrity.

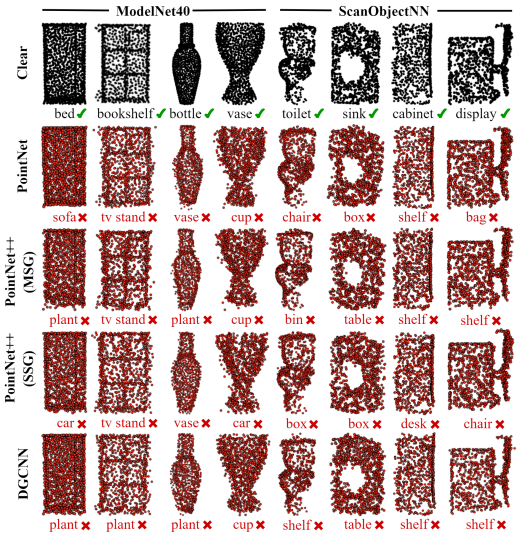

Figure 3: Visualization of adversarial examples.

4.3 Ablation Studies and Other Analysis

Component Effectiveness. Tab. 5 reports ablation results on ModelNet40. After sequentially introducing the Variance Tuning (VT), Normal Constraint (NC), and Momentum (Mom) modules, attack success rates show a continuous upward trend across all target models: VT guides optimization by exploiting local gradient statistics; NC restricts sampling to a reasonable geometric space, ensuring alignment with true distributions for accurate gradient and variance estimation; and Mon enhances stability by accumulating historical gradient information. In summary, the synergy of these modules enables MVT-Attack to achieve highly effective and transferable adversarial attacks.

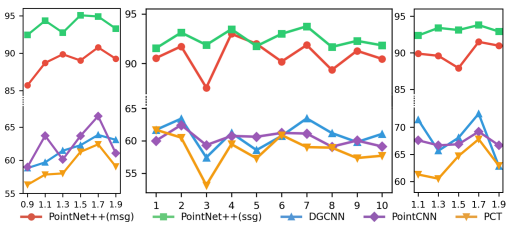

Figure 4: Hyperparameter sensitivity. From left to right: variance coefficient λ, sample size K, and sampling range coefficient β.

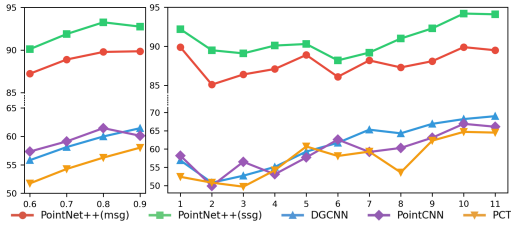

Figure 5: Impact of Momentum Hyperparameters. Left: momentum coefficient μ; Right: momentum window length L.

Sensitivity Analysis of Variance Modulation Hyperparameters We perform a sensitivity analysis using PointNet as the source model on ModelNet40, as shown in Fig. 4. The results indicate that the variance coefficient λ and the sampling range threshold β both achieve optimal performance across target models at 1.7. The sampling number K is relatively insensitive, exhibiting favorable performance at both $K = 2$ and $K = 7$. Given that $K = 7$ yields a marginal overall improvement, we set $\lambda = 1.7$, $K = 7$, and $\beta = 1.7$ for subsequent experiments.

Analysis of Momentum Hyperparameters Fig. 5 illustrates the impact of the momentum coefficient μ and window length L on attack performance. For μ, while both 0.8 and 0.9 yield favorable results, $\mu = 0.9$ demonstrates a distinct advantage on DGCNN and PCT, leading to its selection. Regarding L, performance improves with increasing values and stabilizes around $L = 10$. Consequently, we adopt $\mu = 0.9$ and $L = 10$ for subsequent experiments.

Computational Efficiency and Sampling Strategy Analysis. As shown in Fig. 6, MVT-Attack requires only 0.14 min per sample, achieving superior transferability while maintaining efficiency comparable to PF-Attack (0.10 min). To analyze the impact of sampling strategies, we compare our single-scale sampling (SSS, fixed β) with an enhanced multi-scale sampling (MSS) that estimates variance by adjusting β (see the appendix for details). Results in Fig. 6 (right) show that MSS yields only a marginal 1.0% ASR gain at the cost of a 2.36× computational overhead. Given this unfavorable trade-off, we adopt SSS as the default configuration to balance efficiency and attack performance.

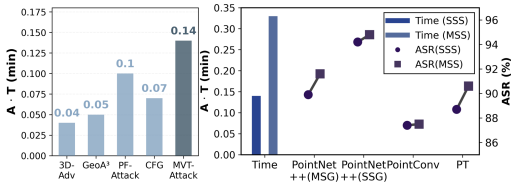

Figure 6: Runtime comparison of attack algorithms (left) and performance comparison of SSS and MSS (right).

Scalability. As shown in Fig. 7, we evaluate the scalability of MVT-Attack (MVT) on the ModelNet40 dataset by integrating it with mainstream transfer attacks (PF-Attack (PF) and CFG) across PointNet and PointNet++(MSG). Integrating MVT consistently improves PF and CFG (e.g., PointNet: PF 42.9%→50.1%, CFG 51.2%→54.7%), verifying the moderate scalability of MVT. In particular, standalone MVT achieves optimal performance (60.5% on PointNet, 64.4% on PointNet++(MSG)), indicating that its core advantage stems from the synergy between a concise loss function and the optimization strategy; introducing complex losses such as PF/CFG disrupts this synergy and leads to performance degradation.

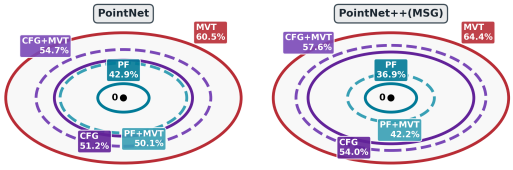

Figure 7: Integration of MVT-Attack into PF-Attack and CFG.

5 Conclusion

In this paper, we propose MVT-Attack, an efficient transferable adversarial attack method for point cloud classification models. By directly rectifying gradients, MVT-Attack improves cross-model attack success rates without relying on complex loss designs, achieving strong performance across mainstream architectures such as PointNet. These results demonstrate that gradient-level optimization provides an effective way to enhance transferability and evaluate the robustness of point cloud classifiers.

AI USE STATEMENT

In this work, we used generative AI tools for drafting sections of the paper and aiding in polishing the writing to improve clarity and flow. We have not used generative AI tools for retrieval and discovery, research ideation or execution, generating synthetic datasets, or proving mathematical claims, and these tasks are not applicable to this work. Additionally, we used generative AI tools for grammar correction and language refinement. We have reviewed all AI-assisted work: all AI-generated text was carefully verified and revised by the authors to ensure factual correctness, logical consistency, and adherence to academic standards. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Jun Chen, Xinke Li, Chongshou Li, and Tianrui Li. Cmfa: Cross-model feature alignment for transferable 3d adversarial attacks. *Neurocomputing*, pp. 133880, 2026.

Xuelong Dai and Bin Xiao. Transferable 3d adversarial shape completion using diffusion models. In *European Conference on Computer Vision*, pp. 392–408. Springer, 2024.

Massimiliano De Sa, Prasanth Kotaru, and Koushil Sreenath. Point cloud-based control barrier function regression for safe and efficient vision-based control. In *2024 IEEE International Conference on Robotics and Automation*, pp. 366–372. IEEE, 2024.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 9185–9193, 2018.

Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7:187–199, 2021.

Abdullah Hamdi, Sara Rojas, Ali Thabet, and Bernard Ghanem. Advpc: Transferable adversarial perturbations on 3d point clouds. In *European Conference on Computer Vision*, pp. 241–257. Springer, 2020.

Bangyan He, Jian Liu, Yiming Li, Siyuan Liang, Jingzhi Li, Xiaojun Jia, and Xiaochun Cao. Generating transferable 3d adversarial point cloud via random perturbation factorization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 764–772, 2023.

Qianjiang Hu, Daizong Liu, and Wei Hu. Exploring the devil in graph spectral domain for 3d point cloud attacks. In *European Conference on Computer Vision*, pp. 229–248. Springer, 2022.

Qidong Huang, Xiaoyi Dong, Dongdong Chen, Hang Zhou, Weiming Zhang, and Nenghai Yu. Shape-invariant 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15335–15344, 2022.

Chongshou Li, Yuheng Liu, Xinke Li, Yuning Zhang, Tianrui Li, and Junsong Yuan. Deep hierarchical learning for 3d semantic segmentation. *International Journal of Computer Vision*, 133(7): 4420–4441, 2025a.

Guanlin Li, Guowen Xu, Han Qiu, Ruan He, Jiwei Li, and Tianwei Zhang. Improving adversarial robustness of 3d point cloud classification models. In *European Conference on Computer Vision*, pp. 672–689. Springer, 2022.

Xinke Li, Chongshou Li, Zekun Tong, Andrew Lim, Junsong Yuan, Yuwei Wu, Jing Tang, and Raymond Huang. Campus3d: A photogrammetry point cloud benchmark for hierarchical understanding of outdoor scene. In *Proceedings of the 28th ACM International Conference on Multimedia*, pp. 238–246, 2020.

Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems*, 31, 2018.

Zezeng Li, Xiaoyu Du, Na Lei, Liming Chen, and Weimin Wang. Nopain: No-box point cloud attack via optimal transport singular boundary. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3492–3502, 2025b.

Hongzhuo Liang, Xiaojian Ma, Shuang Li, Michael Görner, Song Tang, Bin Fang, Fuchun Sun, and Jianwei Zhang. Pointnetgpd: Detecting grasp configurations from point sets. In *2019 IEEE International Conference on Robotics and Automation*, pp. 3629–3635. IEEE, 2019.

Daizong Liu and Wei Hu. Explicitly perceiving and preserving the local geometric structures for 3d point cloud attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3576–3584, 2024.

Tianrui Lou, Xiaojun Jia, Jindong Gu, Li Liu, Siyuan Liang, Bangyan He, and Xiaochun Cao. Hide in thicket: Generating imperceptible and rational adversarial perturbations on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24326–24335, 2024.

Shuchao Pang, Zhenghan Chen, Shen Zhang, Liming Lu, Siyuan Liang, Anan Du, and Yongbin Zhou. Towards a 3d transfer-based black-box attack via critical feature guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 26912–26922, 2025.

Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 652–660, 2017a.

Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 918–927, 2018.

Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30, 2017b.

Keke Tang, Xu He, Weilong Peng, Jianpeng Wu, Yawen Shi, Daizong Liu, Pan Zhou, Wenping Wang, and Zhihong Tian. Manifold constraints for imperceptible adversarial attacks on point clouds. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 5127–5135, 2024.

Keke Tang, Tianyu Hao, Xiaofei Wang, Weilong Peng, Denghui Zhang, Peican Zhu, and Zhihong Tian. Less is more: Sparse and cooperative perturbation for point cloud attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 9430–9438, 2026.

Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 954–962, 2020.

Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1588–1597, 2019.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1924–1933, 2021.

Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019.

Yuxin Wen, Jiehong Lin, Ke Chen, CL Philip Chen, and Kui Jia. Geometry-aware generation of adversarial point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44: 2984–2999, 2020.

Matthew Wicker and Marta Kwiatkowska. Robustness of 3d deep learning in an adversarial setting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11767–11775, 2019.

Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2019.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1912–1920, 2015.

Chong Xiang, Charles R Qi, and Bo Li. Generating 3d adversarial point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9136–9144, 2019.

Tiange Xiang, Chaoyi Zhang, Yang Song, Jianhui Yu, and Weidong Cai. Walk in the cloud: Learning curves for point clouds shape analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 915–924, 2021.

Jiancheng Yang, Qiang Zhang, Rongyao Fang, Bingbing Ni, Jinxian Liu, and Qi Tian. Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*, 2019.

Jianping Zhang, Wenwei Gu, Yizhan Huang, Zhihan Jiang, Weibin Wu, and Michael R Lyu. Curvature-invariant adversarial attacks for 3d point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7142–7150, 2024a.

Jinlai Zhang, Yinpeng Dong, Jun Zhu, Jihong Zhu, Minchi Kuang, and Xiaming Yuan. Improving transferability of 3d adversarial attacks with scale and shear transformations. *Information Sciences*, 662:120245, 2024b.

Renrui Zhang, Liuhui Wang, Ziyu Guo, Yali Wang, Peng Gao, Hongsheng Li, and Jianbo Shi. Parameter is not all you need: Starting from non-parametric networks for 3d point cloud analysis. *arXiv preprint arXiv:2303.08134*, 2023.

Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16259–16268, 2021.

Tianhang Zheng, Changyou Chen, Junsong Yuan, Bo Li, and Kui Ren. Pointcloud saliency maps. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1598–1606, 2019.

Hang Zhou, Kejiang Chen, Weiming Zhang, Han Fang, Wenbo Zhou, and Nenghai Yu. Dup-net: Denoiser and upsampler network for 3d adversarial point clouds defense. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1961–1970, 2019.

Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE/CVF conference on Computer Vision and Pattern Recognition*, pp. 4490–4499, 2018.

APPENDIX

A RESULTS ON DGCNN AND POINTNET++ (SSG)

To further substantiate the superiority of MVT-Attack, we provide supplementary transferability experiments using DGCNN (Wang et al., 2019) and PointNet++ (SSG) (Qi et al., 2017b) as source models. While the main paper primarily reports results based on PointNet (Qi et al., 2017a) and PointNet++ (MSG) (Qi et al., 2017b), extending the evaluation to diverse architectures is essential for establishing the method's universality. As shown in Table 6, regarding the comprehensive

Table 6: Transferability scoreswith DGCNN and PointNet++ (SSG) as source models.

Dataset	ϵ	Transferability Score (%)				
		3D-Adv	GeoA^3	PF-Attack	CFG	**MVT-Attack**
ModelNet40	0.18	7.7	9.1	32.9	45.7	**51.7**
	0.45	7.9	8.9	40.8	58.7	**64.8**
ScanObjectNN	0.18	13.8	16.5	48.5	57.7	**62.1**
	0.45	13.7	16.3	51.5	64.7	**67.2**

Table 7: Attack transferability across classical models on ModelNet40. Best results are in **bold**.

ϵ	Source model	Attack method	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	Point CNN	PCT	Point -PN
0.18	DGCNN	3D-Adv	9.2	11.2	31.2	100.0	3.0	0.4	8.9
		GeoA^3	4.4	27.2	27.6	100.0	8.1	0.0	10.6
		PF-Attack	21.3	60.9	74.8	100.0	29.7	26.7	34.9
		CFG	29.3	70.1	88.9	100.0	45.2	33.5	48.4
		MVT-Attack	**38.2**	**78.1**	**89.0**	100.0	**55.8**	**43.7**	**56.9**
	PointNet++ (SSG)	3D-Adv	7.6	9.6	100.0	6.0	1.3	0.0	4.2
		GeoA^3	5.2	10.4	100.0	2.2	6.0	0.4	7.6
		PF-Attack	13.9	47.4	100.0	19.6	21.8	17.2	26.7
		CFG	18.1	84.7	100.0	38.3	25.5	27.5	39.0
		MVT-Attack	**23.9**	**85.9**	100.0	**41.2**	**34.5**	**32.5**	**41.0**
0.45	DGCNN	3D-Adv	9.6	12.8	30.4	100.0	3.4	1.3	5.9
		GeoA^3	4.4	26.8	25.6	100.0	7.7	0.0	11.0
		PF-Attack	26.1	79.7	85.8	100.0	34.1	37.8	52.9
		CFG	39.0	86.0	91.8	100.0	62.1	44.4	62.9
		MVT-Attack	**43.5**	**93.0**	**95.7**	100.0	**64.9**	**53.5**	**63.7**
	PointNet++ (SSG)	3D-Adv	7.2	10.4	100.0	7.2	1.3	0.0	5.5
		GeoA^3	4.8	9.2	100.0	4.0	5.6	0.4	7.2
		PF-Attack	15.7	56.9	100.0	23.0	24.0	19.4	33.7
		CFG	37.7	86.4	100.0	45.9	39.6	50.5	58.4
		MVT-Attack	**49.4**	**88.2**	100.0	**55.6**	**52.6**	**55.1**	**61.9**

Table 8: Attack transferability across classical models on ScanObjectNN. Best results are in **bold**.

ϵ	Source model	Attack method	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	Point CNN	PCT	Point -PN
0.18	DGCNN	3D-Adv	14.0	17.7	13.4	100.0	14.8	11.3	15.6
		GeoA^3	18.2	21.3	18.6	100.0	19.8	12.8	20.5
		PF-Attack	48.8	58.8	62.9	100.0	60.2	47.3	44.4
		CFG	**58.1**	63.8	**67.7**	100.0	**62.3**	54.0	62.3
		MVT-Attack	57.9	**68.3**	66.6	100.0	59.3	**58.0**	**65.6**
	PointNet++ (SSG)	3D-Adv	15.1	13.9	100.0	16.5	11.3	9.4	12.6
		GeoA^3	17.2	16.6	100.0	13.1	15.3	10.2	14.9
		PF-Attack	47.1	57.1	100.0	34.7	42.2	31.8	47.1
		CFG	50.6	67.8	100.0	48.9	48.7	48.9	58.8
		MVT-Attack	**62.5**	**74.7**	100.0	**55.0**	**61.4**	**53.3**	**62.4**
0.45	DGCNN	3D-Adv	14.4	18.1	13.1	100.0	15.4	11.4	16.5
		GeoA^3	18.4	20.5	18.4	100.0	20.0	12.6	19.8
		PF-Attack	53.7	63.1	65.6	100.0	63.3	47.7	51.0
		CFG	**63.0**	**73.1**	**77.4**	100.0	**69.5**	**62.7**	67.6
		MVT-Attack	58.4	68.7	67.5	100.0	62.2	55.2	**69.0**
	PointNet++ (SSG)	3D-Adv	15.1	13.4	100.0	14.7	11.2	8.6	12.6
		GeoA^3	17.1	16.3	100.0	13.0	15.3	9.4	15.1
		PF-Attack	49.9	61.9	100.0	34.2	42.8	35.4	49.1
		CFG	56.4	76.7	100.0	51.1	53.6	55.9	68.9
		MVT-Attack	**72.7**	**81.0**	100.0	**63.8**	**69.6**	**65.3**	**72.6**

Table 9: Transferability results against PointConv, CurveNet, and PT with PointNet and PointNet++ (MSG) as source models on ModelNet40 and ScanObjectNN. Values denote attack success rates (%); bold indicates the best.

Source model	Attack method	ModelNet40						ScanObjectNN					
		$\epsilon = 0.18$			$\epsilon = 0.45$			$\epsilon = 0.18$			$\epsilon = 0.45$		
		Point Conv	Curve Net	PT	Point Conv	Curve Net	PT	Point Conv	Curve Net	PT	Point Conv	Curve Net	PT
PointNet	3D-Adv	0.8	1.8	0.2	0.8	1.8	1.0	11.6	11.6	13.6	10.9	12.9	12.7
	GeoA^3	10.3	5.5	18.3	14.9	6.7	19.6	29.4	24.8	23.9	26.0	20.8	22.4
	PF-Attack	50.3	22.7	49.9	53.2	24.8	52.3	55.9	39.3	41.6	58.0	45.3	47.5
	CFG	**68.1**	29.2	65.6	81.3	47.5	84.7	57.6	45.8	49.4	60.3	48.2	52.3
	MVT-Attack	64.8	**39.4**	**67.7**	**84.6**	**59.1**	**88.4**	**67.8**	**56.8**	**55.9**	**73.6**	**57.9**	**59.6**
PointNet++ (MSG)	3D-Adv	1.3	2.6	1.4	1.3	2.6	1.8	12.2	8.0	9.8	12.8	8.3	8.8
	GeoA^3	3.8	0.9	7.2	4.1	1.7	7.1	15.6	10.3	13.0	15.2	10.5	11.9
	PF-Attack	50.4	23.1	50.0	52.6	22.9	57.7	57.7	35.6	43.1	59.4	38.4	43.1
	CFG	56.9	26.5	66.0	71.6	46.8	77.6	64.5	53.3	61.8	67.9	66.2	67.5
	MVT-Attack	**63.6**	**31.5**	**68.1**	**80.5**	**55.9**	**84.1**	**69.9**	**58.1**	**75.9**	**77.9**	**69.2**	**80.1**

Table 10: Transferability results against PointConv, CurveNet, and PT with DGCNN and PointNet++ (SSG) as source models on ModelNet40 and ScanObjectNN. Values denote attack success rates (%); bold indicates the best.

Source model	Attack method	ModelNet40						ScanObjectNN					
		$\epsilon = 0.18$			$\epsilon = 0.45$			$\epsilon = 0.18$			$\epsilon = 0.45$		
		Point Conv	Curve Net	PT	Point Conv	Curve Net	PT	Point Conv	Curve Net	PT	Point Conv	Curve Net	PT
DGCNN	3D-Adv	2.9	5.5	4.6	2.9	5.9	8.3	14.4	11.9	16.1	14.5	11.3	12.1
	GeoA^3	9.1	3.8	27.5	11.2	3.8	27.5	20.7	14.7	20.0	20.5	14.7	18.0
	PF-Attack	62.4	23.6	61.7	69.9	35.7	74.8	60.2	44.4	48.1	63.3	51.0	55.8
	CFG	70.1	35.4	75.5	76.5	47.8	85.1	**66.9**	**55.8**	62.3	**73.6**	**64.7**	69.2
	MVT-Attack	**77.2**	**41.2**	**83.5**	**84.6**	**53.6**	**89.3**	62.9	52.7	**76.5**	63.2	55.3	**78.3**
PointNet++ (SSG)	3D-Adv	1.2	0.8	0.8	1.7	0.8	2.9	9.5	7.1	8.0	11.1	7.3	8.6
	GeoA^3	4.1	3.4	2.5	3.3	2.9	3.8	13.5	9.4	10.0	14.7	9.2	11.0
	PF-Attack	38.8	20.2	43.3	43.6	20.3	50.2	55.5	38.6	43.7	59.1	42.0	47.7
	CFG	61.8	31.2	73.3	80.6	49.5	84.8	64.8	54.2	61.2	68.9	61.9	66.5
	MVT-Attack	**68.1**	**35.8**	**74.7**	**84.6**	**59.7**	**87.7**	**72.1**	**59.9**	**71.5**	**78.0**	**70.6**	**77.1**

Transferability Score (TS), MVT-Attack achieves optimal performance across all perturbation budgets ($\epsilon = 0.18$ and $\epsilon = 0.45$) and dataset settings. This consistency with our main findings further corroborates the overall superiority of our approach.

Tables 7 and 8 detail the attack success rates when transferring to specific target models. Overall, MVT-Attack achieves the highest success rate in the vast majority of configurations. Notably, when using PointNet++ (SSG) as the source model, our method exhibits a dominant advantage, outperforming the second-best baseline by substantial margins. The only exception occurs on the ScanObjectNN (Uy et al., 2019) dataset with DGCNN as the source, where our method performs comparably to or slightly lower than the state-of-the-art baseline, CFG (Pang et al., 2025). Nevertheless, excluding this specific scenario, MVT-Attack consistently demonstrates a noticeable lead across different perturbation budgets, fully evidencing its strong cross-model transferability across diverse feature extractors.

B Results on Other Target Models

To further validate the generalizability of MVT-Attack, we extend our evaluation to additional target models, including PointConv (Wu et al., 2019), CurveNet (Xiang et al., 2021), and PT (Zhao et al., 2021). Tables 9 and 10 present detailed transfer attack success rates across four source model

Table 11: Transferability scores evaluated on additional target models (PointConv, CurveNet, and PT).

Dataset	ϵ	Transferability Score (%)				
		3D-Adv	GeoA^3	PF-Attack	CFG	**MVT-Attack**
ModelNet40	0.18	2.0	8.0	41.4	55.0	**59.6**
	0.45	2.7	8.9	46.5	69.5	**76.0**
ScanObjectNN	0.18	11.2	17.1	47.0	58.1	**65.0**
	0.45	10.9	16.2	50.9	63.9	**70.1**

configurations. The results indicate that MVT-Attack achieves the highest performance in the vast majority of scenarios, a trend highly consistent with the conclusions in the main paper and previous supplementary experiments. The only exception occurs on the ScanObjectNN dataset with DGCNN as the source model, where our method performs comparably to or slightly lower than some baselines. This suggests that MVT-Attack maintains robust competitiveness even with minor performance fluctuations in specific configurations. Furthermore, Table 11 summarizes the transferability scores for these additional target models. The data clearly demonstrate that MVT-Attack yields substantial performance improvements over state-of-the-art baselines, achieving an average Transferability Score that exceeds the strongest baseline, CFG, by over 6 across all four scenarios. This substantial margin fully evidences its effectiveness and broad applicability in cross-architecture transfer attacks.

C Formulation of MVT-Attack (MSS)

For notational simplicity, the iteration index t is omitted unless necessary. Let $S = 4$ denote the number of perturbation scales. For each multi-scale neighborhood sampling index $s \in \{0, 1, 2, 3\}$, we associate the corresponding neighborhood with a perturbation radius β_s, where different values of $\beta_s \in \{0.9, 1.1, 1.5, 2.0\}$ represent different neighborhood scales. Neighboring point clouds are generated by sampling perturbations within a local perturbation neighborhood bounded by $\epsilon'_s = \beta_s \cdot \epsilon$, enabling gradient estimation under progressively varying local perturbation ranges. Specifically, at each perturbation scale, K neighboring point clouds are constructed using the normal-constrained sampling procedure described in the main paper. Let $\{G_{s,1}, G_{s,2}, \ldots, G_{s,K}\}$ denote the corresponding gradients computed from these neighboring point clouds. The mean neighborhood gradient at the s-th perturbation scale is then calculated as:

$$\hat{G}_s = \frac{1}{K} \sum_{k=1}^{K} G_{s,k}, \quad s \in \{0, 1, 2, 3\}, \tag{11}$$

let G represents the gradient evaluated at the current adversarial point cloud X_{t-1}^{adv}, which serves as the center of the multi-scale neighborhoods. The scale-specific gradient variance is defined as:

$$V_s = \hat{G}_s - G, \quad s \in \{0, 1, 2, 3\}, \tag{12}$$

where V_s characterizes the gradient fluctuation introduced by neighborhood perturbations at different scales.

Since the gradient is a tensor defined over coordinate channels and point positions, let $M = CN$ denote the total number of gradient elements, where $C = 3$ is the number of coordinate channels, N is the number of points, $c \in \{1, 2, 3\}$ indexes the coordinate channel, and $n \in \{1, \ldots, N\}$ indexes the point position. For every coordinate (c, n), we construct a categorical distribution over the perturbation scales:

$$q_s(c, n) = \frac{|V_s(c, n)|^2 + p}{\sum_{i=0}^{S-1} (|V_i(c, n)|^2 + p)}, \quad s \in \{0, \ldots, S-1\}, \tag{13}$$

where $p > 0$ is a small constant for numerical stability. By construction, $\sum_{s=0}^{S-1} q_s(c, n) = 1$.

We then define the average relative contribution of the s-th scale as:

$$P_s = \frac{1}{M} \sum_c \sum_n q_s(c, n). \tag{14}$$

To characterize the ambiguity of the scale contribution, we further define the scale-specific contribution to the normalized cross-scale entropy:

$$H_s = -\frac{1}{M \log S} \sum_c \sum_n q_s(c, n) \log q_s(c, n). \tag{15}$$

A reliable scale should provide a sufficiently large average contribution while exhibiting low cross-scale ambiguity. Accordingly, we define the scale logit and its normalized weight as:

$$z_s = \log(P_s + p) - H_s, \tag{16}$$

$$w_s = \frac{\exp(z_s)}{\sum_{i=0}^{S-1} \exp(z_i)}. \tag{17}$$

The final multi-scale gradient-deviation term is obtained by:

$$\widehat{V} = \sum_{s=0}^{S-1} w_s V_s. \tag{18}$$

Finally, $\widehat{V}_t$ replaces the single-scale deviation V_t in the gradient-rectification step:

$$G_t'' = G_t + \lambda \frac{\widehat{V}_t}{\|\widehat{V}_t\|_1 + p}. \tag{19}$$

D Evaluation of MVT-Attack (MSS)

To further investigate the effectiveness of the multi-scale sampling strategy (MSS), we conduct additional cross-architecture transferability evaluations with diverse source and target backbones. Specifically, besides the source models used in the main experiments, we additionally include DGCNN and PointNet++(SSG) as source architectures to provide a more comprehensive assessment. We adopt the Transferability Score (TS), which averages the attack success rates across all unseen target models, to quantitatively measure the overall transferability performance. As reported in Tab. 12, MVT-Attack achieves a TS of 53.4%, while the MSS variant further improves this score to 54.1%, outperforming all compared baselines. This result demonstrates that multi-scale variance estimation can further enhance the transferability of adversarial examples, although the improvement is relatively marginal.

The detailed cross-model transfer results are presented in Tables 13 and 14. Compared with existing approaches, both MVT-Attack and MVT-Attack (MSS) achieve consistently stronger transferability across different source-target architecture combinations. In particular, MSS provides additional

Table 12: Transferability scores of various attacks (including MVT-Attack (MSS)) across all models on ModelNet40.

ϵ	Transferability Score (%)					
	3D-Adv	GeoA^3	PF-Attack	CFG	**MVT-Attack**	**MVT-Attack (MSS)**
0.18	5.5	8.3	35.6	47.3	53.4	**54.1**

Table 13: Attack success rates (%) on ModelNet40 under $\epsilon = 0.18$: classical target models. Bold values indicate the best transfer results.

Source	Method	PointNet	PointNet++ (MSG)	PointNet++ (SSG)	DGCNN	Point CNN	PCT	Point-PN
PointNet	3D-Adv	100.0	8.4	10.4	6.8	1.0	0.0	1.2
	GeoA^3	100.0	20.0	19.6	7.2	2.1	3.3	5.5
	PF-Attack	100.0	49.3	61.5	24.8	29.5	23.4	26.1
	CFG	100.0	72.6	80.6	39.3	32.7	30.4	42.5
	MVT-Attack	100.0	73.5	84.0	**46.8**	46.0	**41.4**	48.5
	MVT-Attack (MSS)	100.0	**74.8**	**84.9**	45.3	**47.4**	40.8	**49.0**
PointNet++ (MSG)	3D-Adv	6.8	100.0	28.4	11.2	1.0	1.8	5.7
	GeoA^3	4.4	100.0	14.4	6.4	4.4	0.0	6.9
	PF-Attack	17.2	100.0	67.0	24.3	22.2	16.6	28.9
	CFG	17.6	100.0	72.9	25.3	21.0	26.4	35.1
	MVT-Attack	26.1	100.0	75.0	33.1	**36.6**	33.8	43.0
	MVT-Attack (MSS)	**29.3**	100.0	**75.6**	**38.5**	34.3	**34.5**	**45.2**
DGCNN	3D-Adv	9.2	11.2	31.2	100.0	3.0	0.4	8.9
	GeoA^3	4.4	27.2	27.6	100.0	8.1	0.0	10.6
	PF-Attack	21.3	60.9	74.8	100.0	29.7	26.7	34.9
	CFG	29.3	70.1	88.9	100.0	45.2	33.5	48.4
	MVT-Attack	**38.2**	78.1	**89.0**	100.0	55.8	43.7	**56.9**
	MVT-Attack (MSS)	33.3	**79.5**	**89.0**	100.0	**57.6**	**44.1**	54.8
PointNet++ (SSG)	3D-Adv	7.6	9.6	100.0	6.0	1.3	0.0	4.2
	GeoA^3	5.2	10.4	100.0	2.2	6.0	0.4	7.6
	PF-Attack	13.9	47.4	100.0	19.6	21.8	17.2	26.7
	CFG	18.1	84.7	100.0	38.3	25.5	27.5	39.0
	MVT-Attack	23.9	**85.9**	100.0	41.2	34.5	32.5	41.0
	MVT-Attack (MSS)	**24.4**	85.5	100.0	**42.7**	**34.9**	**33.2**	**41.6**

Table 14: Attack success rates (%) on ModelNet40 under $\epsilon = 0.18$: additional target models. Bold values indicate the best transfer results.

Source	Method	PointConv	CurveNet	PT
PointNet	3D-Adv	0.8	1.8	0.2
	GeoA^3	10.3	5.5	18.3
	PF-Attack	50.3	22.7	49.9
	CFG	68.1	29.2	65.6
	MVT-Attack	64.8	**39.4**	67.7
	MVT-Attack (MSS)	**72.0**	37.1	**76.9**
PointNet++ (MSG)	3D-Adv	1.3	2.6	1.4
	GeoA^3	3.8	0.9	7.2
	PF-Attack	50.4	23.1	50.0
	CFG	56.9	26.5	66.0
	MVT-Attack	**63.6**	31.5	68.1
	MVT-Attack (MSS)	60.1	**32.8**	**69.0**
DGCNN	3D-Adv	2.9	5.5	4.6
	GeoA^3	9.1	3.8	27.5
	PF-Attack	62.4	23.6	61.7
	CFG	70.1	35.4	75.5
	MVT-Attack	**77.2**	**41.2**	83.5
	MVT-Attack (MSS)	75.6	40.7	**86.9**
PointNet++ (SSG)	3D-Adv	1.2	0.8	0.8
	GeoA^3	4.1	3.4	2.5
	PF-Attack	38.8	20.2	43.3
	CFG	61.8	31.2	73.3
	MVT-Attack	**68.1**	**35.8**	**74.7**
	MVT-Attack (MSS)	65.3	34.5	**74.7**

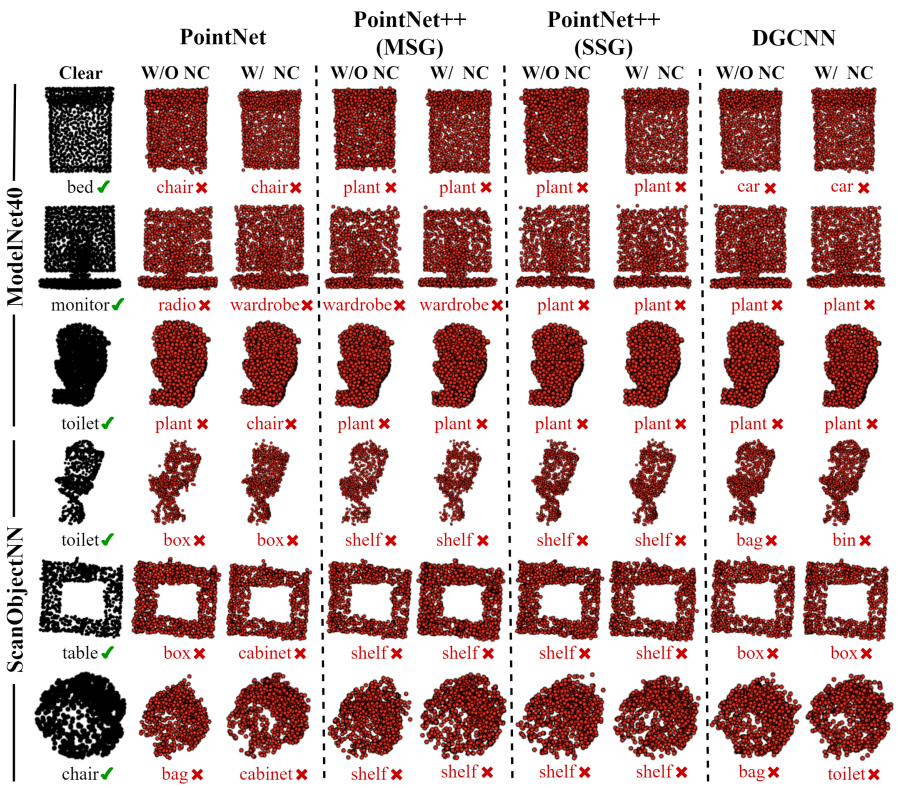

Figure 8: Qualitative comparison of adversarial point clouds generated with the proposed Normal Constraint (NC) (W/ NC) and without NC (W/O NC). NC generally preserves geometric structures more faithfully and produces less perceptible perturbations.

improvements in many transfer scenarios, such as transfers from PointNet to PointCNN, PT, and from DGCNN to PointCNN and PT, indicating that multi-scale sampling helps capture more generalizable perturbation patterns beyond a single source model. These observations further support our analysis in the main paper that the variance modulation strategy alleviates source-model overfitting and improves the robustness of generated adversarial perturbations. Considering the limited performance gain and additional computational cost introduced by MSS, we select the single-scale sampling strategy as the default configuration to achieve a better balance between attack effectiveness and efficiency.

E Additional Visualization of Normal-Constrained Sampling

To further evaluate the effectiveness of the proposed Normal Constraint (NC), we provide additional qualitative comparisons between adversarial point clouds generated with and without NC. Specifically, representative adversarial examples generated by four source models, namely PointNet, PointNet++ (MSG), PointNet++ (SSG), and DGCNN, are visualized on both ModelNet40 and ScanObjectNN.

As shown in Fig. 8, adversarial point clouds generated with NC generally preserve the original geometric structures more faithfully than those generated without NC. Across the majority of examples, NC effectively alleviates irregular point deviations and local geometric distortions, resulting in smoother object surfaces and less perceptible perturbations. These observations are consistently reflected across different source models and datasets.

The qualitative results suggest the effectiveness of the proposed NC strategy. By constraining neighborhood sampling along local surface normals, NC facilitates more reliable gradient variance estimation, thereby improving the transferability of adversarial examples. Meanwhile, the resulting perturbations better conform to the underlying object geometry, leading to improved imperceptibility.

F PSEUDOCODE OF MVT-ATTACK

Algorithm 1 presents the complete pseudocode of MVT-Attack, which iteratively generates adversarial point clouds via variance-driven gradient rectification and exponentially decaying momentum aggregation under normal constraints.

Algorithm 1 MVT-Attack

Input: benign Point cloud X, true label y, source model f, iteration steps T, step size η, perturbation threshold ϵ, hyperparameters K, λ, μ, L
Output: Adversarial point cloud X_T^{adv}

Initialize perturbation $\Delta_0 \sim \mathcal{N}(0, 0.001)$;
Initialize momentum buffer $\mathcal{H} \leftarrow [\mathbf{0}]_{i=1}^{L}$;
$X_0^{\text{adv}} \leftarrow X + \Delta_0$;
for $t = 1$ **to** T **do**
  $G_t \leftarrow \nabla_{X_{t-1}^{\text{adv}}} \mathcal{L}(f(X_{t-1}^{\text{adv}}), y)$; \\ current gradient
  Perform Normal-Constrained Sampling around X_{t-1}^{adv};
  Compute sampled gradients $\{G'_{t,1}, \ldots, G'_{t,K}\}$;
  Estimate gradient variance $V_t \leftarrow \frac{1}{K} \sum_{j=1}^{K} G'_{t,j} - G_t$;
  Rectify gradient $G''_t \leftarrow G_t + \lambda \cdot \dfrac{V_t}{\|V_t\|_1 + p}$;
  Update momentum buffer $\mathcal{H} \leftarrow [G''_t] \oplus \mathcal{H}$;
  if $|\mathcal{H}| > L$ **then**
    Keep the latest L gradients;
  end if
  Aggregate momentum $\bar{G}_t \leftarrow \frac{\sum_{i=0}^{min(L-1,t-1)} \mu^{i+1} \mathcal{H}[i]}{\sum_{i=0}^{min(L-1,t-1)} \mu^{i+1}}$;
  Update perturbation $\Delta_t \leftarrow \Delta_{t-1} - \eta \cdot \bar{G}_t$;
  Project each point-wise perturbation in Δ_t onto its corresponding surface normal;
  if $\|\Delta_t\|_2 > \epsilon$ **then**
    $\Delta_t \leftarrow \epsilon \cdot \dfrac{\Delta_t}{\|\Delta_t\|_2}$;
  end if
  $X_t^{\text{adv}} \leftarrow X + \Delta_t$;
end for
return X_T^{adv}