Infinite Babble: Inflating 3D Vision-Language Model Inference Overhead via Adversarial Geometric Perturbation

Shuoyang Sun[1†], Jiaxin Hong[1†], Yv Zhang[1†], Kuofeng Gao[2], Hao Fang[2], Fan Mo[3], Bin Chen[1*], Shu-Tao Xia[2]
[1]Harbin Institute of Technology, Shenzhen
[2]Tsinghua Shenzhen International Graduate School, Tsinghua University
[3]Huawei Technology

Abstract

3D Vision-Language Models (3D-VLMs) have emerged as the critical cognitive backbone for spatial intelligence, enabling precise reasoning over unstructured 3D data. While these models serve as the foundation for downstream robotics and embodied systems, their reliance on autoregressive decoding introduces a fundamental vulnerability regarding inference efficiency. In this work, we present **Inflate3D**, a novel adversarial framework designed to trigger computational and economic exhaustion in 3D-VLMs. Specifically, we exploit the model's sensitivity to untrusted 3D assets to hijack the generation process. Inflate3D operates by injecting imperceptible noise that forces the model into a state of pathological verbosity, effectively stalling the inference pipeline. Our approach comprises two synergistic strategies: (1) a *semantic-aware adversarial manipulation* that leverages internal representations to selectively perturb semantically critical regions while preserving geometric structure, and (2) a *trajectory disruption mechanism* that manipulates token probabilities to suppress End-of-Sequence (EOS) emission, thereby prolonging decoding and inducing verbose outputs. Experiments on standard benchmarks show that Inflate3D amplifies output length and energy consumption by up to **6.45**×, demonstrating a potent capability to drain system resources. These findings expose a critical blind spot in multimodal alignment, highlighting the urgent need to secure spatial foundation models against resource exhaustion attacks.

1 Introduction

Recent advances in 3D Vision-Language Models (3D-VLMs) such as PointLLM (Xu et al., 2024, 2025) and X-InstructBLIP (Panagopoulou et al., 2023) have significantly improved spatial-language reasoning, enabling robust performance in a wide range of 3D tasks including object understanding, scene-level decision-making, and multimodal navigation. These models typically adopt a language foundation model (LLM) as their backbone and extend it to the 3D domain by aligning textual and unstructured geometric modalities, thereby generalizing the LLM's linguistic capabilities to visual and spatial perception. Consequently, 3D-VLMs are increasingly deployed in high-stakes applications such as autonomous driving, embodied AI, and digital reconstruction.

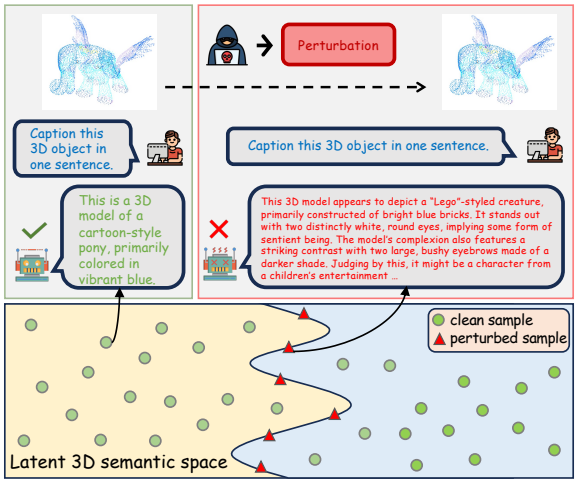


Figure 1: Demonstration of Inflate3D's effects. **Top (output level):** Adversarial perturbations significantly prolong auto-regressive decoding while preserving input fidelity. **Bottom (model level):** Perturbations shift samples into regions of semantic ambiguity, blurring distinctions from clean examples in semantic space.

However, the powerful generative and reasoning capabilities of 3D-VLMs come at the cost of enormous model sizes and expensive inference pipelines. In particular, their reliance on autoregressive decoding requires substantial computational resources per query. Moreover, due to the limited availability and high cost of constructing 3D datasets, users often rely on publicly available 3D

† Equal contribution.
* Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 5249–5267
July 2-7, 2026 ©2026 Association for Computational Linguistics

assets (e.g., meshes, point clouds) sourced from the internet to support downstream reasoning tasks. This dependency creates a critical attack surface: an adversary may subtly manipulate these untrusted 3D inputs to hijack the decoding process. By inducing 3D-VLMs to generate pathologically long outputs, attackers can trigger a resource exhaustion attack. Crucially, this vulnerability manifests as a double-edged sword: **it imposes prohibitive latency and battery drain on local deployments (e.g., robotics), while maliciously inflating token billing for cloud-based API services.**

Several prior works have explored resource exhaustion attacks in text and image modalities. In text, Sponge examples (Shumailov et al., 2021) increase inference cost by maximizing activation norms. Engorgio Prompt (Dong et al., 2024) demonstrates that crafted prompts can suppress end-of-sequence (EOS) token generation in LLMs, forcing abnormally long outputs without degrading semantic quality. In the vision-language setting, NICGSlowdown (Chen et al., 2022) manipulates logit dynamics to delay EOS emission, and Verbose Images (Gao et al., 2024a) introduce imperceptible perturbations that promote diverse outputs, amplifying computational overhead.

Despite these efforts, existing computation-based attacks have primarily focused on LLMs and 2D-VLMs, whereas the study of 3D-VLMs remains largely unexplored. The unique characteristics of 3D domains pose distinctive challenges for effective attacks, such as maintaining geometric fidelity under perturbation, ensuring robust cross-modal alignment, and addressing the semantic sparsity inherent in 3D point clouds. To bridge this gap, we present **Inflate3D**, the first adversarial framework that performs semantic-aware resource exhaustion attacks against 3D-VLMs through purely 3D-modal perturbations. Inflate3D injects imperceptible but strategically crafted noise into point cloud inputs to hijack the model's generation process and inflate inference overhead. The key components of our method are two-fold: (1) *Semantic-aware adversarial manipulation*: We leverage the model's internal hidden states to identify semantically important tokens and selectively perturb their corresponding points in the input space, disrupting the model's reasoning process while preserving the overall geometric structure. (2) *Trajectory disruption mechanism*: We maintain high-entropy token predictions and suppress premature EOS emission to prolong auto-regressive decoding, triggering pathological verbosity and significantly increasing inference-time energy consumption.

Figure 1 illustrates the disruptive effect of Inflate3D. Experiments on widely-used 3D-VLM benchmarks show that Inflate3D substantially increases inference decoding steps and energy consumption, exposing a critical vulnerability in 3D multimodal reasoning systems. By synergistically combining semantic-aware adversarial manipulation and a trajectory disruption mechanism, the attack induces excessively long output sequences in a highly stealthy and resource-draining manner.

Our contributions are summarized as follows:

- To the best of our knowledge, we propose the first attack framework named Inflate3D, which performs semantic-aware resource exhaustion attacks on 3D-VLMs via imperceptible perturbations crafted entirely in the 3D input space.
- We design a novel trajectory disruption mechanism that effectively manipulates token dispersion and suppresses EOS emission to prolong decoding, inducing verbose and energy-expensive outputs.
- We conduct comprehensive experiments on the Objaverse (Deitke et al., 2023) and ModelNet40 (Wu et al., 2015) benchmarks, showing that Inflate3D increases output length and energy cost by up to 6.45× and 6.12× respectively, demonstrating consistent and powerful attack performance.

2 Related Work

2.1 Recent Advances in 3D-VLMs

Current 3D Vision-Language Models (3D-VLMs) typically follow two main paradigms: multi-view rendering and direct point-cloud alignment. The first category leverages 2D VLMs by rendering 3D objects into multi-view images. Notably, 3D-LLM (Hong et al., 2023) employs CLIP-like encoders (Radford et al., 2021) and BLIP (Li et al., 2023) to bridge 3D perception with language reasoning. Conversely, point-cloud-based methods align 3D data directly with LLMs. Point-Bind (Guo et al., 2023) constructs a joint embedding space via ImageBind (Girdhar et al., 2023), enabling generation through models like ImageBind-LLM (Han et al., 2023). PointLLM (Xu et al., 2024, 2025) and LEO (Huang et al., 2023) encode point clouds

into latent tokens for auto-regressive processing, facilitating end-to-end spatial reasoning.

Recent works further enhance architectural scalability and fine-grained scene understanding. ShapeLLM (Qi et al., 2024) and X-InstructBLIP (Panagopoulou et al., 2023) focus on robust cross-modal alignment for effective zero-shot generalization. Computational efficiency and scalability are addressed by MiniGPT-3D (Tang et al., 2024) and GreenPLM (Tang et al., 2025), with the latter achieving highly data-efficient learning using only 12% of typical training data volume. For complex or dynamic scenes, Video-3D LLM (Zheng et al., 2025) incorporates 3D positional encoding into video representations, while LSceneLLM (Zhi et al., 2025) targets large-scale scene understanding via adaptive preference identification mechanisms.

2.2 Resource Exhaustion Attacks

While traditional Denial-of-Service (DoS) attacks target availability via massive request floods (Elleithy et al., 2005; Aldhyani and Alkahtani, 2023; Bhatia et al., 2018; Mirkovic and Reiher, 2004; Long and Thomas, 2001), the rise of adaptive neural networks has introduced algorithmic resource exhaustion threats. Unlike fixed-cost networks, modern models exhibit input-dependent computation, making them vulnerable to adversarial inputs that trigger excessive latency or energy consumption (Hong et al., 2020; Liu et al., 2023). Early works such as sponge samples (Shumailov et al., 2021) and NICGSlowDown (Chen et al., 2022) exploit this by maximizing activation norms or manipulating end-of-sequence (EOS) logits.

In the generative era, these attacks have evolved to target LLMs and VLMs. Engorgio Prompt (Dong et al., 2024) and Verbose Images (Gao et al., 2024a) manipulate textual or visual inputs to suppress EOS generation, forcing the model to produce abnormally long sequences, while VLMInferSlow (Wang et al., 2025) broadens this threat to realistic black-box ML-as-a-service settings. Furthermore, (Gao et al., 2024b) extend energy-latency manipulation to the video domain by encouraging diverse temporal features. Similarly, LLMEffiChecker (Feng et al., 2024) systematically identifies tokens that degrade inference efficiency. In the 3D domain, Poison-splat (Lu et al., 2024) increases the rendering overhead of 3D Gaussian Splatting (Kerbl et al., 2023) via multi-view perturbations. Despite these advances, **resource exhaustion attacks targeting 3D-VLMs remain largely unexplored**. As 3D-VLMs combine complex geometric reasoning with cross-modal generation, they present unique vulnerabilities distinct from pure LLMs or 2D VLMs. Our work systematically investigates these risks to guide the development of robust and efficient 3D architectures.

3 Methodology

3.1 Method Overview

To extend generated sequences while keeping perturbations imperceptible, Inflate3D leverages the intrinsic 3D point cloud structure through two synergistic modules. Semantic-aware Adversarial Manipulation identifies semantically critical points based on token importance and selectively perturbs them to preserve local geometry. Trajectory Disruption Mechanism prolongs generated outputs using a *dispersion loss* to increase token uncertainty and a *persistence loss* to delay EOS emission. To balance the two losses, we adopt a projection-based adjustment optimization method. Together, these components generate longer outputs with minimal geometric distortion, amplifying the inference-time cost of 3D-VLMs. An overview of the attack pipeline is shown in Figure 2.

3.2 Problem Formulation

3D-VLM Inference Procedure. We consider a 3D-VLM composed of an encoder f_{enc} and an auto-regressive decoder f_{dec}. Let $P \in \mathbb{R}^{N\times 3}$ denote the input point cloud and $T = \{t_1, \ldots, t_M\}$ the textual instruction. The encoder maps P into latent point tokens H_p and the textual input into tokens H_t. These are concatenated to form the initial context for the decoder. The decoder then generates an output sequence $Y = \{y_1, \ldots, y_S\}$ auto-regressively. At decoding step s, the model predicts token y_s conditioned on the encoded inputs and the prefix $Y_{<s} = \{y_1, \ldots, y_{s-1}\}$. Concretely, the decoder logits at step s are:

$$\mathbf{z}_s(P, T, Y_{<s}) = f_{\text{dec}}\big([H_p; H_t], Y_{<s}\big), \quad (1)$$

and the conditional probability of generating token $v \in V$ at step s is:

$$p(v \mid P, T, Y_{<s}) = \text{softmax}\big(\mathbf{z}_s(P, T, Y_{<s})\big)_v. \quad (2)$$

Attack Objective. We aim to craft an imperceptible perturbation δ applied to the point cloud P

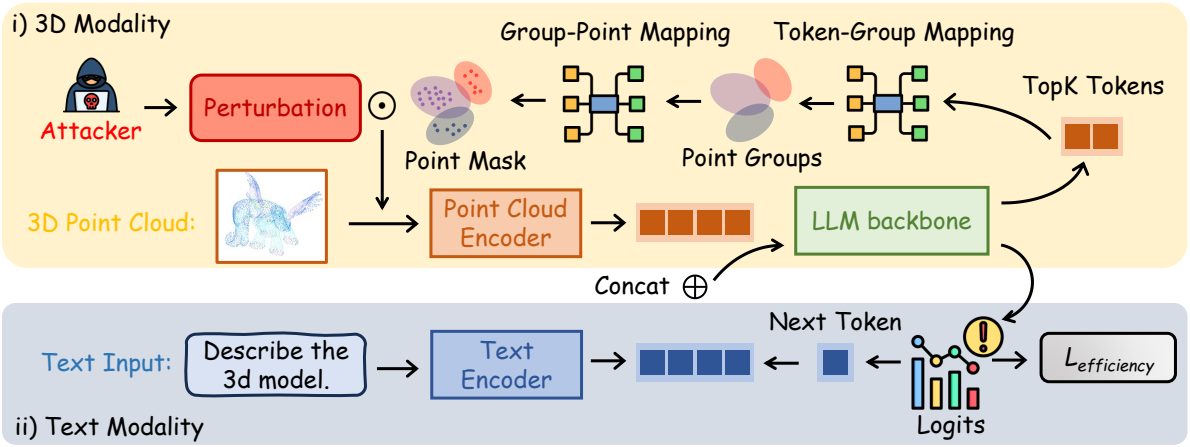


Figure 2: Overall pipeline of Inflate3D. The input 3D point cloud P is first tokenized, and subtle perturbations are injected only at semantically critical points indicated via the token–group–point mapping, creating targeted semantic ambiguity. Point tokens are then combined with textual embeddings to drive auto-regressive decoding in the LLM. By evaluating the predicted logits, the efficiency-oriented loss $\mathcal{L}_{\text{efficiency}}$ directs the perturbations to extend output sequences and amplify computational cost.

to obtain a perturbed input $\tilde{P} = P + \delta$, such that the 3D-VLM generates abnormally long output sequences Y, increasing inference-time computation. This is formalized by minimizing an efficiency-oriented objective:

$$\min_{\|\delta\|_\infty \le \epsilon} \mathcal{L}_{\text{efficiency}}\big(f_{\text{dec}}([f_{\text{enc}}(P+\delta); H_t])\big), \quad (3)$$

where ϵ bounds the perturbation magnitude. The perturbation δ should preserve the geometric integrity of the original point cloud (See details in Section 3.3) while the minimization of $\mathcal{L}_{\text{efficiency}}$ aims to reduce the decoding efficiency, thereby inducing high energy consumption during inference (See details in Section 3.4).

3.3 Semantic-aware Adversarial Manipulation

To preserve the geometric fidelity of the input, we introduce a semantic-aware adversarial manipulation strategy that perturbs only a subset of semantically critical points. This strategy consists of three steps: (1) estimating token importance, (2) tracing vulnerabilities via Token-Group and Group-Point mappings, and (3) applying masked perturbations under perceptual constraints.

We first measure the importance of each point token using its decoder hidden states $\{\mathbf{h}_j^{(\ell)}\}_{j=1}^n$. Specifically, we focus on the final layer $\ell = L$, where activations directly contribute to the next-token prediction logits. The top-k tokens are then selected according to a ratio ρ:

$$\mathcal{I} = \text{TopK}\big(\{\|\mathbf{h}_j^{(L)}\|_2\}_{j=1}^n,\ k = \lfloor \rho n \rfloor\big). \quad (4)$$

This step highlights the tokens with the strongest influence on the next-token logits and therefore the model's generation behavior. To rigorously define the reverse tracing process (as explicitly visually decomposed in Figure 2), we formulate two sequential mappings: the **Token-Group Mapping** ($\mathcal{F}_{\text{T}\to\text{G}}$) and the **Group-Point Mapping** ($\mathcal{F}_{\text{G}\to\text{P}}$).

First, the Token-Group mapping associates each critical token index $j \in \mathcal{I}$ back to its corresponding local group of point indices, formally defined as $\mathcal{F}_{\text{T}\to\text{G}}(j) = \mathcal{G}_j$, which is originally established during the encoder's forward spatial grouping stage. Subsequently, the Group-Point mapping $\mathcal{F}_{\text{G}\to\text{P}}$ projects these index groups into the actual 3D coordinate space by extracting the corresponding raw points. By applying this mapping across all critical tokens, we extract the final semantically critical point set $\mathcal{P}$:

$$\mathcal{P} = \bigcup_{j \in \mathcal{I}} \mathcal{F}_{\text{G}\to\text{P}}(\mathcal{G}_j). \quad (5)$$

In practice, we construct a **semantic-aware mask** $M \in \{0,1\}^N$, rigorously defined by the indicator function $M_i = \mathbb{I}(P_i \in \mathcal{P})$, to mark these points, ensuring that gradients outside $\mathcal{P}$ are suppressed. Finally, the adversarial perturbation is applied via additive coordinate shifts in masked form:

$$\tilde{P} = P + \delta \odot M, \quad (6)$$

where δ denotes the additive perturbation constrained by an ℓ_∞ budget to maintain imperceptibility. During optimization, only masked points are updated, focusing perturbations on semantically influential regions while leaving the majority of the point cloud untouched. This semantic-aware mask design enables strong attack effectiveness with minimal perceptual distortion.

3.4 Trajectory Disruption Mechanism

Building upon the semantic-aware perturbation strategy described above to maintain imperceptibility, we further introduce our trajectory disruption mechanism. It aims to prolong the auto-regressive decoding trajectory, which maximizes the inference-time energy cost of 3D-VLMs.

From Adversarial Examples to Semantic Ambiguity. In classification tasks, adversarial examples (AEs) are known to probe the decision boundary by applying small perturbations that induce misclassification (He et al., 2018; Ilyas et al., 2019; Rice et al., 2020). However, large language models, particularly 3D-VLMs, operate in an open-ended generative setting rather than a closed label space. In this context, the notion of a decision boundary must be reinterpreted. Instead of forcing high-confidence misclassifications, we transfer the functionality of AEs into the semantic space of generative models: perturbations are designed to push samples toward regions of high uncertainty, where token predictions become more ambiguous.

Definition of SAI. Formally, let $\tilde{P} = P + \delta$ denote the perturbed point cloud, T the input text, and Y the generated output sequence. Let $\hat{\mathbf{z}}_s \in \mathbb{R}^{|V|}$ denote the model's logits for the s-th token, and define

$$\hat{p}_s(\cdot) := \mathrm{softmax}(\hat{\mathbf{z}}_s), \tag{7}$$

where $\hat{p}_s(v)$ gives the predicted probability for token $v \in V$. We call $\tilde{P}$ a **spatially ambiguous instance (SAI)** if its predictive distribution exhibits sufficiently high entropy:

$$\mathcal{H}(\hat{p}_s) \geq \log|V| - \tau, \tag{8}$$

where $\mathcal{H}(\hat{p}_s) = -\sum_{v \in V} \hat{p}_s(v) \log \hat{p}_s(v)$, $\log|V|$ is the entropy of the uniform distribution over the vocabulary, and $\tau \geq 0$ is a slack threshold. Intuitively, a smaller τ enforces a distribution closer to uniform (higher ambiguity); in the extreme $\tau = 0$ the condition requires near-uniformity. SAIs therefore correspond to perturbed samples that lie near the semantic boundary of the model's latent space, producing ambiguous predictions that prolong auto-regressive decoding and inflate inference-time energy consumption.

Dispersion loss. This geometric interpretation provides intuition for why SAIs exhibit unstable decoding behavior, and it naturally motivates the need for an objective that can deliberately induce such high-entropy states. To actively construct SAIs, we define a dispersion loss that encourages the predicted distributions to approach the maximum entropy distribution across the full sequence:

$$\mathcal{L}_{\mathrm{d}} = \mathbb{E}_{s=1}^{S}\Big[\mathrm{KL}(\hat{p}_s \parallel \mathcal{U})\Big], \tag{9}$$

where $\mathcal{U}$ is the uniform distribution over the vocabulary V, and the expectation is taken over sequence positions $s = 1, \ldots, S$. Minimizing $\mathcal{L}_{\mathrm{d}}$ drives the predicted distributions toward high entropy, thereby encouraging SAIs as defined in Eq. 8.

Persistence loss. Entropy maximization alone only diversifies token predictions but leaves the stopping behavior uncontrolled. To regulate the termination dynamics, we design a persistence loss that discourages premature EOS emission:

$$\mathcal{L}_{\mathrm{p}} = \mathbb{E}_{s=1}^{S}\Big[\hat{p}_s\big(\mathrm{EOS} \mid \tilde{P}, T, Y_{<s}\big)\Big], \tag{10}$$

where $\hat{p}_s\big(\mathrm{EOS} \mid \tilde{P}, T, Y_{<s}\big)$ denotes the predicted EOS probability at position s. This penalizes early EOS predictions and enforces a more persistent decoding trajectory. By combining the dispersion and persistence terms, we obtain the overall efficiency-oriented objective:

$$\mathcal{L}_{\text{efficiency}} = \mathcal{L}_{\mathrm{d}} + \mathcal{L}_{\mathrm{p}}. \tag{11}$$

This joint formulation drives predictions toward high-entropy distributions while simultaneously suppressing EOS emission, thereby prolonging auto-regressive trajectories and inflating inference-time energy consumption in 3D-VLMs.

Gradient conflict optimization. When jointly optimizing $\mathcal{L}_{\mathrm{d}}$ and $\mathcal{L}_{\mathrm{p}}$, their gradients may conflict. To stabilize the optimization, we adopt a projection-based adjustment inspired by PCGrad (Yu et al., 2020). Given gradients g_{d} and g_{p}, if their inner product is negative, we remove the conflicting component:

$$g_{\mathrm{d}} \leftarrow g_{\mathrm{d}} - \frac{\langle g_{\mathrm{d}}, g_{\mathrm{p}} \rangle}{\|g_{\mathrm{p}}\|^2 + \sigma}\, g_{\mathrm{p}}, \tag{12}$$

Table 1: Quantitative comparison of different resource exhaustion attacks on four target models across Objaverse and ModelNet40 datasets. Best results are highlighted in **bold**.

Method	PointLLM			X-InstructBLIP			MiniGPT-3D			GreenPLM		
	Length	Latency	Energy	Length	Latency	Energy	Length	Latency	Energy	Length	Latency	Energy
(a) Dataset: Objaverse												
Original	19.77	0.84	66.45	16.87	1.76	98.65	27.84	**6.92**	183.96	16.52	0.92	43.03
Gaussian Noise	20.58	1.08	104.78	11.86	1.22	72.68	12.98	6.91	181.70	16.25	1.04	45.84
Random Drop	25.27	1.32	127.95	16.62	1.72	100.35	26.28	6.87	190.14	17.67	1.09	47.80
PGD	34.77	3.20	143.72	13.07	0.73	37.91	29.57	6.34	186.94	14.36	1.02	61.4
NICGSlowdown	29.99	1.46	119.16	12.82	0.74	43.80	26.25	6.76	141.33	15.97	1.24	44.26
Inflate3D	**127.52**	**5.65**	**406.50**	**39.03**	**2.83**	**272.43**	**74.71**	6.79	**553.37**	**41.48**	**1.52**	**67.10**
(b) Dataset: ModelNet40												
Original	14.76	0.65	52.45	10.04	1.62	88.52	11.34	**6.93**	239.10	13.69	0.82	14.01
Gaussian Noise	22.84	0.89	70.62	7.91	1.67	92.34	13.00	6.92	243.54	16.69	0.91	15.26
Random Drop	19.26	0.78	62.32	10.12	1.62	89.29	11.34	6.92	237.48	13.93	0.84	14.46
PGD	24.19	1.78	79.74	9.16	0.90	63.25	23.52	6.70	193.54	12.24	0.94	52.21
NICGSlowdown	18.19	0.95	19.75	9.38	0.96	56.80	28.92	6.68	136.93	15.70	1.18	45.91
Inflate3D	**45.90**	**1.88**	**139.27**	**32.85**	**2.40**	**193.54**	**65.87**	6.71	**383.17**	**35.58**	**1.50**	**59.82**

where σ is a small constant for numerical stability. The final update direction is then:

$$g_{\text{efficiency}} = g_{\text{d}} + g_{\text{p}}. \tag{13}$$

This adjustment mitigates destructive interference and improves optimization stability.

4 Experiments

4.1 Experimental Configurations

Datasets and Models. We conduct experiments on two point-cloud datasets: Objaverse (Deitke et al., 2023), a large-scale collection of diverse 3D objects with rich annotations, and ModelNet40 (Wu et al., 2015), a widely used benchmark of CAD-based 3D models. For each dataset, we uniformly sample 8,192 points per object and randomly select 100 point clouds as the evaluation set. For 3D-VLMs, we consider four representative models that take raw point clouds as input: PointLLM (Xu et al., 2024), X-InstructBLIP (Panagopoulou et al., 2023), GreenPLM (Tang et al., 2025), and MiniGPT-3D (Tang et al., 2024), ensuring that the evaluation focuses purely on point-cloud understanding without interference from auxiliary modalities.

Comparison Baselines. To the best of our knowledge, we are the first to investigate resource exhaustion attacks against 3D-VLMs. Given the absence of prior work in this specific domain, we establish a benchmark by adapting representative methods to the 3D-VLM setting. We compare Inflate3D against two categories of baselines: (1) Common Corruptions: Gaussian Noise and Random Drop, which simulate realistic point cloud imperfections; (2) Adversarial Attacks: PGD (Madry et al., 2017) and NICGSlowDown (Chen et al., 2022), which we modified to operate on 3D inputs. Comparisons against these diverse baselines highlight the superior effectiveness of our method in maximizing inference costs compared to general corruptions or adapted adversarial strategies.

Evaluation Metrics. Our evaluation considers efficiency-oriented metrics, including average output sequence length, inference latency (s), and energy consumption (J), which reflect the ability of an attack to exhaust computational resources.

Implementation Details. For all models, the default prompts are applied, and the maximum output length is set to 2,048 tokens. To ensure a fair comparison, Inflate3D and the adapted adversarial baselines (PGD and NICGSlowDown) are all optimized for 100 iterations with the same perturbation bound of $\ell_\infty \leq 0.1$. Gaussian Noise is also applied with the same $\ell_\infty \leq 0.1$ bound, while Random Drop directly removes 10% of the points. The mask ratio ρ of Inflate3D is set to 50%. Additional architectural settings and implementation details are provided in Appendix A.

4.2 Main Results

Table 1 presents a quantitative comparison of inference overheads across original samples, natural corruptions, adapted adversarial attacks, and our proposed Inflate3D.

The original samples serve as a baseline. Natural corruptions and adapted adversarial baselines

Table 2: Robustness of Inflate3D against representative defense pipelines on Objaverse and ModelNet40.

Setting	Objaverse			ModelNet40		
	Length	Latency	Energy	Length	Latency	Energy
Original	19.77	0.84	66.45	14.76	0.65	52.45
Inflate3D	127.52	5.65	406.50	45.90	1.88	139.27
Inflate3D+Dup-Net	62.62	1.56	363.21	23.20	0.63	143.75
Inflate3D+SOR	62.52	1.54	354.68	24.29	0.66	150.17
Inflate3D+SRS	64.50	1.60	371.94	23.88	0.65	148.11

Table 3: Transferability of Inflate3D from PointLLM to different target models on Objaverse and ModelNet40.

Target Model	Objaverse			ModelNet40		
	Length	Latency	Energy	Length	Latency	Energy
MiniGPT-3D	70.06	6.70	582.79	61.01	6.70	616.14
GreenPLM	46.51	1.21	51.49	59.48	1.42	65.87
X-InstructBLIP	19.38	0.59	77.70	12.84	0.52	111.16

(PGD, NICGSlowDown) yield significantly weaker performance, with metrics exhibiting only marginal fluctuations. We attribute this to a misalignment with the *cross-modal generation dynamics*: baselines lack the **semantic steering** necessary to effectively compel 3D-VLMs into extended generation. In stark contrast, Inflate3D demonstrates clearly superior effectiveness. By explicitly maximizing model uncertainty and increasing the entropy of token-probability distributions, our method fundamentally disrupts the *visual-semantic alignment*. This forces the model into a state of high perplexity, triggering verbose corrections or repetitive loops. Consequently, PointLLM's response length on Objaverse inflates from $\sim$19.77 to 127.52 tokens (a 6.45$\times$ increase), causing a 512% surge in energy consumption, highlighting a critical and previously overlooked efficiency vulnerability.

Regarding model robustness, we observe a distinct hierarchy clearly linked to model capacity. Large-scale models (PointLLM, MiniGPT-3D) are consistently the most fragile under attack. Conversely, GreenPLM remains relatively robust (costs rise only $\sim 2\times$). We attribute this to its lightweight architecture, which imposes an intrinsic bias towards brevity. Its limited decoding capacity acts as a natural bottleneck, effectively preventing the generation of sustained long-context sequences even under strong perturbations.

To gain further interpretability, we visualize the learned embeddings via t-SNE on 100 randomly selected samples from the ModelNet40 dataset (Figure 3). Results reveal a favorable trade-off: at the instance level, the global layout remains largely stable, preserving visual imperceptibility; however, at the token level, clusters become highly concentrated. This reflects a drastic alteration in token-level semantics, effectively decoupling the visual input from the intended concise text representation to drive resource escalation. For a more comprehensive assessment, the detailed distributions of efficiency-oriented metrics are provided in Appendix B.

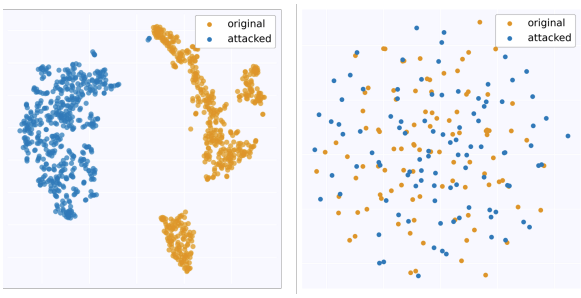


Figure 3: T-SNE visualization of learned embeddings under Inflate3D. Left: token-level embeddings become more compact. Right: instance-level embeddings retain the overall layout.

4.3 Robustness and Transferability

Robustness against Defense Pipelines. Real-world applications often employ preprocessing to filter outliers. Therefore, we evaluate the robustness of Inflate3D against three representative point cloud defense methods: **SRS, SOR, and Dup-Net**, following the implementation protocols in (Zhou et al., 2019). As shown in Table 2, while these defenses introduce a moderate reduction in attack strength, our method remains highly effective. Taking the Objaverse dataset as a representative example, the defenses reduce the sequence length from $\sim$127 to $\sim$63 tokens. Crucially, even after such purification, the generated sequence remains over 3$\times$ longer than the clean baseline ($\sim$20), leading to a corresponding energy consumption surge of over 5$\times$. This resilience indicates that the perturbations optimized by Inflate3D are not merely removable high-frequency noise. Instead, our method alters semantically critical regions of the point cloud structure that persist even after defense, demonstrating robust efficacy against standard countermeasures. Furthermore, we also demonstrate Inflate3D's resilience against output-side defenses, specifically the decoding repetition penalty, in Appendix C.

Transferability to Black-box Models. To assess whether Inflate3D poses a threat beyond the white-box setting, we evaluate its transferability by generating adversarial samples using PointLLM as the

Table 4: Ablation of loss objectives.

$\mathcal{L}_{\mathrm{d}}$	$\mathcal{L}_{\mathrm{p}}$	PCGrad	Length	Latency	Energy
✓			101.85	4.62	320.79
	✓		31.42	1.53	116.79
✓	✓		124.25	5.21	336.17
✓	✓	✓	127.52	5.65	406.50

Table 5: Ablation on mask method.

Method	Length	Latency	Energy
Original	14.76	0.65	52.45
Random Mask	42.85	2.25	148.46
Ours	45.90	1.88	139.27

source model and transferring them to three target models. The results are reported in Table 3. We observe that the attack maintains strong performance on structurally similar models. Specifically, on ModelNet40, the transferred samples successfully induce MiniGPT-3D and GreenPLM to generate significantly extended sequences (∼61 tokens), imposing substantial inference overheads compared to clean inputs. We attribute this high transferability to the shared architectural vulnerabilities in the 3D-LLM backbones. In contrast, the attack is less effective on X-InstructBLIP (length remains low at ∼12-19 tokens). This limited transferability aligns with its distinct cross-modal alignment mechanism, confirming that architectural differences can act as a barrier to black-box transfer.

4.4 Ablation Study and Further Analysis

Effect of Loss Objectives. We dissect the contributions of each optimization objective in Table 4, taking PointLLM on Objaverse as the primary testbed. The results indicate that the dispersion loss ($\mathcal{L}_{\mathrm{d}}$) primarily drives resource escalation. By forcing token prediction probabilities toward a uniform distribution, $\mathcal{L}_{\mathrm{d}}$ maximizes output entropy, driving the model into a high-uncertainty state that yields prolonged sequences. In parallel, the persistence loss ($\mathcal{L}_{\mathrm{p}}$) acts as a critical constraint by suppressing EOS probability to prevent premature termination. However, directly summing these objectives reveals a bottleneck: their gradient landscapes often conflict, where increasing entropy may inadvertently counteract the suppression of termination signals. Integrating PCGrad successfully resolves this multi-objective dilemma by projecting conflicting gradients onto the normal plane of each other. This structural alignment ensures that

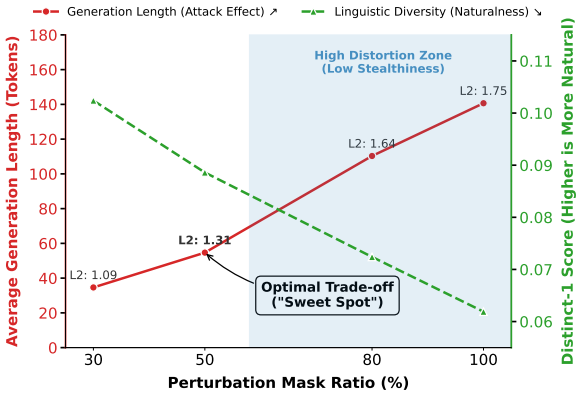


Figure 4: Trade-off analysis on perturbation mask ratios. The plot highlights the conflict between attack effectiveness (**Red**: Generation Length) and linguistic quality (**Green**: Distinct-1 Score). Geometric distortion (L_2 Norm) is annotated on the curve. The **50% ratio** emerges as the optimal "Sweet Spot," maximizing impact while avoiding the severe distortion (shaded) and mode collapse seen at higher ratios.

both objectives are optimized synergistically rather than competitively. Consequently, this combination unlocks the highest attack severity, peaking at 127.52 tokens and 406.50 J, confirming that harmonizing dispersion and persistence is essential for maximizing adversarial efficiency.

Effect of Masking Strategy. We further validate the necessity of our semantic-aware mask against a random masking baseline on ModelNet40 (Table 5). Even random masking induces notable overhead, reflecting the structural vulnerability of 3D-LLMs to unstructured input noise. However, our semantic-aware strategy achieves superior performance, reaching a sequence length of 45.90 compared to the baseline. This performance margin stems from precision: unlike blind disruption, Inflate3D leverages gradient guidance to identify and perturb semantically critical regions—those geometric features holding the highest causal weight in visual token formation. By concentrating distortion on these high-value areas, we ensure that every perturbed point contributes maximally to disrupting visual-text alignment, achieving a more efficient attack per unit of distortion. A comprehensive analysis justifying this masking necessity—detailing its critical role in preventing linguistic mode collapse and its superiority over random masking in complex environments (e.g., ScanQA (Azuma et al., 2022))—is provided in Appendix D.

Effect of Perturbation Ratio. We investigate the trade-off between attack potency and imperceptibil-

ity by varying the mask ratio ρ on ModelNet40 using 100 randomly selected samples (Figure 4). We monitor three dimensions: generation length (resource consumption), feature L_2 norm (geometric deviation), and Distinct-1 (linguistic naturalness). As the ratio increases, a distinct divergence is observed: while attack potency escalates, it incurs severe penalties in imperceptibility. Pushing the ratio beyond 50% drives the feature space into a high-distortion regime, accompanied by linguistic mode collapse. In this state, the Distinct-1 score plummets, indicating that the output degenerates into unnatural repetitive loops easily detectable by defenders (qualitative failure modes are further illustrated in Appendix D). In contrast, the 50% ratio emerges as the sweet spot. It secures a substantial attack impact while preserving linguistic diversity and high geometric fidelity—quantitatively verified by an extremely low Chamfer Distance of 2.6×10^{-4} (detailed in Appendix D). This confirms that targeted perturbation is sufficient to trigger expensive semantic loops without crossing into perceptible distortion or structural breakdown. Additional ablation studies are detailed in Appendix E.

Further Qualitative Analysis. Beyond the quantitative metrics and ablation studies presented above, we provide extensive qualitative examples across diverse 3D-VLM architectures to demonstrate the universality of the induced verbose generation in Appendix F. Additionally, detailed visualizations of our semantic-aware mask targets are provided in Appendix G.

5 Conclusion

In this work, we presented Inflate3D, the first adversarial framework for semantic-aware resource exhaustion attacks against 3D-VLMs via 3D perturbations. By integrating semantic-aware manipulation with trajectory disruption, Inflate3D induces verbose decoding and substantially increases inference energy consumption while remaining imperceptible to human observers. Experiments across multiple datasets, point cloud resolutions, and masking strategies demonstrate effectiveness and generalizability, revealing a critical vulnerability in 3D multimodal reasoning systems and motivating future work on efficiency-aware defenses.

Limitations

While Inflate3D effectively exposes 3D-VLM efficiency vulnerabilities, several limitations remain. First, semantic-aware adversarial manipulation requires iterative gradient backpropagation to optimize the perturbation mask. Unlike inference-only attacks, this process introduces additional latency during generation, which may constrain its applicability in strictly real-time scenarios where rapid injection is required.

Second, our method currently operates under a white-box assumption, using internal gradients to compute the efficiency-oriented losses. Although we demonstrate transferability, the attack success rate diminishes on architectures with distinct cross-modal alignment mechanisms, indicating that robust black-box transferability remains an open challenge.

Finally, consistent with prior studies, our evaluation is confined to English prompts and standard 3D datasets. Given the complexity of spatial reasoning across linguistic contexts, extending this mechanism to multilingual 3D-VLMs remains an avenue for future work to validate broader generalizability.

Ethical Statement

This paper presents a novel adversarial framework to advance the safety and robustness of 3D multimodal systems. We acknowledge that Inflate3D, capable of triggering high energy consumption and latency, could be misused for Denial-of-Service (DoS) attacks against cloud APIs or battery-constrained robots. However, our goal is not to equip adversaries but to expose efficiency blind spots in current 3D-VLMs, motivating more robust architectures.

To ensure safety, resource exhaustion experiments were conducted in controlled laboratory environments using isolated hardware to avoid impacting public services. Furthermore, by demonstrating how imperceptible 3D perturbations hijack decoding trajectories, this work highlights the urgent need for efficiency-aware defenses. We preliminarily evaluated defenses like Dup-Net and SOR to encourage stronger countermeasures. All datasets and models used are open-source and utilized in compliance with their licenses.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under grant 62301189, 62576122, 62571298, Guangdong Basic and Applied Basic Research Foundation under grant 2026A1515011139.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

THH Aldhyani and H Alkahtani. 2023. Cyber security for detecting distributed denial of service attacks in agriculture 4.0: Deep learning model. mathematics, 11 (233).

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. 2022. Scanqa: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19129–19139.

Sajal Bhatia, Sunny Behal, and Irfan Ahmed. 2018. Distributed denial of service attacks and defense mechanisms: current landscape and future directions. In *Versatile Cybersecurity*, pages 55–97. Springer.

Simin Chen, Zihe Song, Mirazul Haque, Cong Liu, and Wei Yang. 2022. Nicgslowdown: Evaluating the efficiency robustness of neural image caption generation models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15365–15374.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, and 1 others. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153.

Jianshuo Dong, Ziyuan Zhang, Qingjie Zhang, Tianwei Zhang, Hao Wang, Hewu Li, Qi Li, Chao Zhang, Ke Xu, and Han Qiu. 2024. An engorgio prompt makes large language model babble on. *arXiv preprint arXiv:2412.19394*.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Khaled M Elleithy, Drazen Blagovic, Wang K Cheng, and Paul Sideleau. 2005. Denial of service attack techniques: analysis, implementation and comparison.

Xiaoning Feng, Xiaohong Han, Simin Chen, and Wei Yang. 2024. Llmeffichecker: Understanding and testing efficiency degradation of large language models. *ACM Transactions on Software Engineering and Methodology*, 33(7):1–38.

Kuofeng Gao, Yang Bai, Jindong Gu, Shu-Tao Xia, Philip Torr, Zhifeng Li, and Wei Liu. 2024a. Inducing high energy-latency of large vision-language models with verbose images. *arXiv preprint arXiv:2401.11170*.

Kuofeng Gao, Jindong Gu, Yang Bai, Shu-Tao Xia, Philip Torr, Wei Liu, and Zhifeng Li. 2024b. Energy-latency manipulation of multi-modal large language models via verbose samples. *arXiv preprint arXiv:2404.16557*.

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15180–15190.

Ziyu Guo, Renrui Zhang, Xiangyang Zhu, Yiwen Tang, Xianzheng Ma, Jiaming Han, Kexin Chen, Peng Gao, Xianzhi Li, Hongsheng Li, and 1 others. 2023. Point-bind & point-llm: Aligning point cloud with multi-modality for 3d understanding, generation, and instruction following. *arXiv preprint arXiv:2309.00615*.

Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, and 1 others. 2023. Imagebind-llm: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*.

Warren He, Bo Li, and Dawn Song. 2018. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*.

Sanghyun Hong, Yiğitcan Kaya, Ionuţ-Vlad Modoranu, and Tudor Dumitraş. 2020. A panda? no, it's a sloth: Slowdown attacks on adaptive multi-exit neural network inference. *arXiv preprint arXiv:2010.02432*.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 2023. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. 2023. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. 2019. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Sebastien Bubeck, Caio César Teodoro Mendes, Weizhu Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, and 1 others. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Han Liu, Yuhao Wu, Zhiyuan Yu, Yevgeniy Vorobeychik, and Ning Zhang. 2023. Slowlidar: Increasing the latency of lidar-based detection using adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5146–5155.

Neil Long and Rob Thomas. 2001. Trends in denial of service attack technology. *CERT Coordination Center*, 648(651):569.

Jiahao Lu, Yifan Zhang, Qiuhong Shen, Xinchao Wang, and Shuicheng Yan. 2024. Poison-splat: Computation cost attack on 3d gaussian splatting. *arXiv preprint arXiv:2410.08190*.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Jelena Mirkovic and Peter Reiher. 2004. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53.

Artemis Panagopoulou, Le Xue, Ning Yu, Junnan Li, Dongxu Li, Shafiq Joty, Ran Xu, Silvio Savarese, Caiming Xiong, and Juan Carlos Niebles. 2023. X-instructblip: A framework for aligning x-modal instruction-aware representations to llms and emergent cross-modal reasoning. *arXiv preprint arXiv:2311.18799*.

Zekun Qi, Runpei Dong, Shaochen Zhang, Haoran Geng, Chunrui Han, Zheng Ge, Li Yi, and Kaisheng Ma. 2024. Shapellm: Universal 3d object understanding for embodied interaction. In *European Conference on Computer Vision*, pages 214–238. Springer.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, and 1 others. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR.

Leslie Rice, Eric Wong, and Zico Kolter. 2020. Overfitting in adversarially robust deep learning. In *International conference on machine learning*, pages 8093–8104. PMLR.

Ilia Shumailov, Yiren Zhao, Daniel Bates, Nicolas Papernot, Robert Mullins, and Ross Anderson. 2021. Sponge examples: Energy-latency attacks on neural networks. In *2021 IEEE European symposium on security and privacy (EuroS&P)*, pages 212–231. IEEE.

Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. 2023. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*.

Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Yixue Hao, Long Hu, and Min Chen. 2024. Minigpt-3d: Efficiently aligning 3d point clouds with large language models using 2d priors. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 6617–6626.

Yuan Tang, Xu Han, Xianzhi Li, Qiao Yu, Jinfeng Xu, Yixue Hao, Long Hu, and Min Chen. 2025. More text, less point: Towards 3d data-efficient point-language understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 7284–7292.

Xiasi Wang, Tianliang Yao, Simin Chen, Runqi Wang, Lei Ye, Kuofeng Gao, Yi Huang, and Yuan Yao. 2025. Vlminferslow: Evaluating the efficiency robustness of large vision-language models as a service. In *ACL*.

Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 2015. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. 2024. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pages 131–147. Springer.

Runsen Xu, Shuai Yang, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. 2025. Pointllm-v2: Empowering large language models to better understand point clouds. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Le Xue, Ning Yu, Shu Zhang, Artemis Panagopoulou, Junnan Li, Roberto Martín-Martín, Jiajun Wu, Caiming Xiong, Ran Xu, Juan Carlos Niebles, and 1 others. 2024. Ulip-2: Towards scalable multimodal pre-training for 3d understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 27091–27101.

Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. 2020. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33:5824–5836.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. 2022. Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19313–19322.

Duo Zheng, Shijia Huang, and Liwei Wang. 2025. Video-3d llm: Learning position-aware video representation for 3d scene understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8995–9006.

Hongyan Zhi, Peihao Chen, Junyan Li, Shuailei Ma, Xinyu Sun, Tianhang Xiang, Yinjie Lei, Mingkui Tan, and Chuang Gan. 2025. Lscenellm: Enhancing large 3d scene understanding using adaptive visual preferences. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 3761–3771.

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

Junsheng Zhou, Jinsheng Wang, Baorui Ma, Yu-Shen Liu, Tiejun Huang, and Xinlong Wang. 2023. Uni3d: Exploring unified 3d representation at scale. *arXiv preprint arXiv:2310.06773*.

A More Experimental Details

In this work, we focus on **3D point clouds** as the input modality (instead of projecting them into images or multi-view representations). Accordingly, we adopt several open-source and competitive 3D-to-text models as baselines. Below we describe their architectural details and the common implementation protocols.

PointLLM Settings. The adopted PointLLM (Xu et al., 2024) is composed of a Point-BERT (Yu et al., 2022) encoder pre-trained with ULIP-2 (Xue et al., 2024), a multi-layer projector with GeLU activations to map point features into the embedding space, and a Vicuna-7B (Chiang et al., 2023) LLM as the language backbone. With two additional special tokens, the vocabulary size is 32003. Prompt template used in PointLLM is `"Describe the 3d model."`.

X-InstructBLIP Settings. The adopted X-InstructBLIP (Panagopoulou et al., 2023) integrates a Point-BERT (Yu et al., 2022) encoder pre-trained with ULIP-2 (Xue et al., 2024) for 3D representation learning, a Q-Former module that learns query tokens, and a Vicuna-7B (Chiang et al., 2023) as the language backbone. All Q-Formers are initialized with BLIP-2 (Li et al., 2023) stage-1 weights to ensure stable training and effective multimodal alignment. Prompt template used in X-InstructBLIP is `"Describe the 3d model."`.

MiniGPT-3D Settings. The adopted MiniGPT-3D (Tang et al., 2024) is composed of a Q-Former initialized from BLIP-2 (Li et al., 2023), a Mixture of Query Experts (MQE) to enhance semantic representation, a modality projector for aligning point queries with the text embedding space, and the Phi-2-2.7B backbone (Javaheripi et al., 2023) as the language model. Prompt template used in MiniGPT-3D is `"Caption the object in short."`.

GreenPLM Settings. The adopted GreenPLM (Tang et al., 2025) consists of a ViT (Dosovitskiy et al., 2020) point encoder and an EVA-CLIP-E (Sun et al., 2023) text encoder, both trained by Uni3D (Zhou et al., 2023), a two-layer MLP projector with GeLU activation to align encoder outputs with the language embedding space, and the Phi-3 (Abdin et al., 2024) model as the language backbone. Prompt template used in GreenPLM is :`"Caption this 3D model in detail."`.

Implementation Notes

- **General Settings:** All experiments are conducted with FP16 precision. Input point clouds are uniformly sampled to $8,192$ points, except for specific ablation studies on point density.
- **Model Configuration:** All models are evaluated in their default inference mode without any fine-tuning. For ModelNet40, objects are assigned a fixed black color to compensate for missing texture.
- **Hardware & Metrics:** Efficiency metrics (latency and energy) are measured on a single NVIDIA H20 GPU and averaged over multiple runs to ensure reliability.

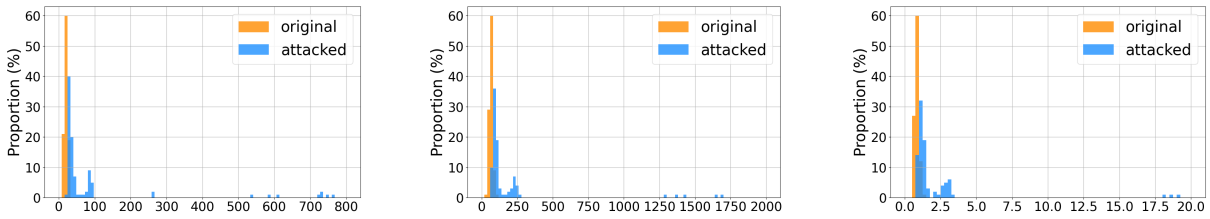
B Distribution of Efficiency-oriented Metrics

We visualize the empirical distributions of output length, latency, and energy in Figure 5. Results exhibit a pronounced bimodal distribution rather than a uniform shift. The primary mode aligns with clean baselines, representing mild perturbations. However, a secondary high-magnitude mode emerges at the tail, corresponding to successful "jailbreaks" trapped in pathological loops. This suggests a threshold effect: once perturbations cross a latent tipping point, they trigger self-reinforcing verbose decoding. For real-world deployments, this variance creates severe instability. Unlike predictable linear costs, the stochastic nature of these outliers makes resource provisioning challenging, as a small fraction of queries can unexpectedly monopolize system capacity.

C Evaluation against Output-Side Defenses

To evaluate the robustness of our attack against output-side defenses, we conducted experiments incorporating a repetition penalty during decoding. Specifically, we applied a 1.1 penalty, a common configuration to suppress repetitive and degenerate text generation.

Most resource exhaustion attacks rely on forcing pathological repetitions of few tokens. Such patterns are inherently fragile and easily neutralized by repetition penalties that significantly penalize recurring tokens. In contrast, Inflate3D circumvents this limitation through its semantic-aware masking



(a) Length (b) Energy (c) Latency

Figure 5: Distribution of PointLLM results on Objaverse: (a) Sequence Length, (b) Energy Consumption, and (c) Inference Latency.

Table 6: Robustness of Inflate3D against output-side defense. RP denotes Repetition Penalty (set to 1.1).

Setting	Objaverse			ModelNet40		
	Length	Latency	Energy	Length	Latency	Energy
Original	19.77	0.84	66.45	14.76	0.65	52.45
Inflate3D	127.52	5.65	406.50	45.90	1.88	139.27
Inflate3D + RP	94.08	4.57	321.74	59.21	2.58	191.31

Table 7: Ablation study of masking strategies on ScanQA dataset.

Masking Strategy	Length	Latency	Energy
Original	22.30	1.20	107.48
Random Mask	69.25	4.63	431.67
Ours	91.64	6.67	633.82

strategy. By perturbing semantically critical 3D regions, we achieve an optimal trade-off between visual imperceptibility and attack efficacy. Instead of low-entropy repetitions, our approach misleads the 3D-VLM into generating diverse, plausible, and excessively long descriptions (detailed in Appendix D).

As shown in Table 6, Inflate3D consistently induces a substantial increase in resource consumption despite the penalty constraint. While the defense partially mitigates effects on Objaverse, the resulting sequence length and energy costs remain significantly higher than the baseline. Interestingly, on ModelNet40, the penalty actually exacerbated the attack, leading to even higher overheads. This suggests that when forced to avoid repetitions, the compromised LLM can be driven to generate even longer, more convoluted sequences. Overall, these results demonstrate that our semantic-aware perturbations are highly robust against common output-side countermeasures.

D Necessity of Semantic-Aware Masking

To justify the design of our semantic-aware mask, we first analyze the failure modes of an unconstrained attack (i.e., 100% perturbation). As hypothesized, blindly perturbing the entire point cloud fundamentally destroys the intrinsic geometric semantics required for coherent visual-language alignment. Figure 6 illustrates the severely degraded model outputs under 100% masking. In this high-distortion regime, the 3D-VLM suffers from complete representational collapse, producing highly repetitive sequences of single letters or function words. While technically "long," such sequences are linguistically unnatural and easily neutralized by standard repetition penalties. Thus, the semantic-aware mask is a critical constraint to maintain linguistic plausibility and evade automated anomaly detection.

Comparison with Random Masking in Complex Scenes. Beyond avoiding total linguistic collapse, the semantic-aware strategy is essential for distinguishing our attack from naive random perturbations. We observe a performance saturation on datasets with simple, object-centric geometries (e.g., ModelNet40). In such confined cases, even unstructured random perturbations have a predictably high statistical probability of covering the few critical regions, ultimately leading to a marginal performance gap between the two masking strategies.

However, the superiority of our approach becomes evident in complex environments. As shown in Table 7, we conducted an ablation study on the ScanQA dataset, which involves large-scale, intricate 3D scenes with extensive background clutter. In these challenging scenarios, random masking struggles to induce significant overhead because it fails to consistently target the most influential regions for visual-language reasoning. In contrast, our Inflate3D significantly outperforms random masking by precisely locating and perturbing only the most semantically critical points. These results confirm that our semantic-aware strategy is a nec-

Table 8: Geometric fidelity across masking ratios.

Masking Ratio (%)	CD ($\times 10^{-4}$)
30%	1.8
50% (Ours)	2.6
80%	4.0
100%	4.7

Table 9: Ablation study on the input point number.

Points	Objaverse			ModelNet40		
	Length	Latency	Energy	Length	Latency	Energy
2048	52.10	2.15	154.04	46.38	2.44	179.23
4096	140.71	7.71	503.01	45.91	1.88	138.54
8192	127.52	5.65	406.50	45.90	1.88	139.27

essary solution for real-world, complex 3D-VLM applications.

Quantitative Analysis of Geometric Fidelity. To further strengthen our claims regarding imperceptibility, we employ the Chamfer Distance (CD) to quantitatively evaluate geometric fidelity. As shown in Table 8, there is an inherent trade-off between the perturbation scale and geometric fidelity; as the masking ratio increases, the CD value naturally rises. However, our empirically chosen 50% ratio serves as an optimal balance. As further corroborated by Figure 4, this 50% ratio emerges as a "sweet spot." It maximizes the attack's impact while successfully evading the severe linguistic mode collapse associated with larger masks. Consequently, it triggers significant resource exhaustion while preserving visual integrity and keeping the CD extremely low at 2.6×10^{-4}. This confirms that our semantic-aware perturbations are both functionally effective and highly stealthy.

E Ablation Study on Point Cloud Size

We investigate the sensitivity of Inflate3D to input resolution by varying the point count $N \in \{2048, 4096, 8192\}$ on PointLLM, as summarized in Table 9.

On **Objaverse**, we observe distinct degradation at 2,048 points, where sequence length drops from 127.52 (8k) to 52.10 tokens. This suggests that fine-grained objects require sufficient density to maintain the semantic integrity necessary for perturbation. Interestingly, performance peaks at 4,096 (140.71 tokens), indicating an optimal trade-off between semantic representation and perceptibility.

In contrast, **ModelNet40** remains robust across all resolutions, stabilizing around $\sim$46 tokens regardless of density (46.38 at 2k vs. 45.90 at 8k). We attribute this to the dataset's simplified CAD nature. Consequently, even sparse representations retain core semantic features to trigger resource exhaustion, confirming the attack's versatility.

F Additional Qualitative Examples

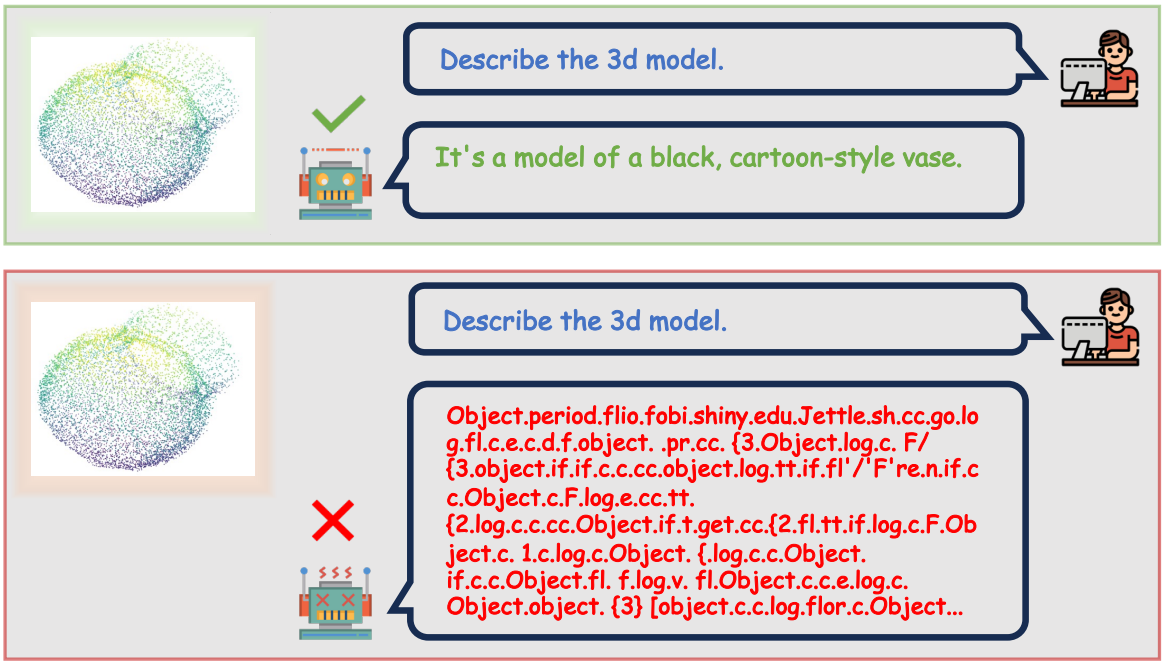
In this section, we provide additional qualitative visualization results to supplement the main experiments. Figures 7 through 10 present the attack outcomes on four representative 3D-VLMs: PointLLM, X-InstructBLIP, MiniGPT-3D, and GreenPLM, respectively. All examples utilize samples from the Objaverse dataset. These visualizations further demonstrate the universality of Inflate3D, showing its capability to consistently induce verbose generation and computational overhead across diverse model architectures.

G Visualization of Our Semantic-aware Mask

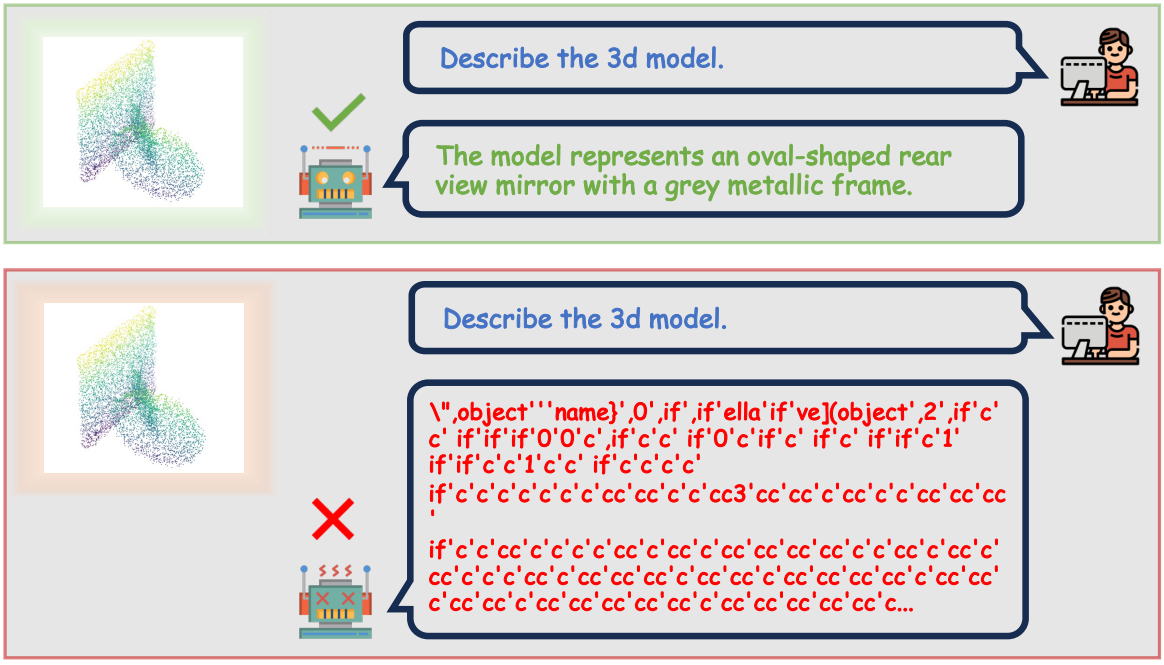
We visualize our semantic-aware mask on Objaverse (Deitke et al., 2023) (Figure 11) and ModelNet40 (Wu et al., 2015) (Figure 12). Top rows display the applied mask highlighting perturbation targets, while bottom rows show original references. As observed, the mask consistently targets core regions strongly associated with semantic meaning. For instance, it prioritizes characteristic object parts in Objaverse and salient geometries—like chair legs or airplane wings—in ModelNet40. This confirms that Inflate3D concentrates perturbations on semantically critical areas most influential for model understanding, bypassing irrelevant points.

H LLM usage

We used an OpenAI LLM (GPT-5) as a writing and formatting assistant. In particular, it helped refine grammar and phrasing, improve textual flow and clarity, and suggest edits to figure/table captions and layout (e.g., column alignment, caption length, placement). The LLM did not contribute to research ideation, experimental design, implementation, data analysis, or technical content beyond surface-level edits. All outputs were carefully reviewed and edited by the authors, who take full responsibility for the final text and visuals.



(a) Failure Case 1



(b) Failure Case 2

Figure 6: Qualitative examples of attacks on ModelNet40 using PointLLM with a **100% perturbation ratio** (without semantic-aware masking). Excessive distortion causes the model to collapse into obvious loops of repetitive characters or words. Such outputs, while long, are easily detectable by defenders due to their extremely low linguistic naturalness.

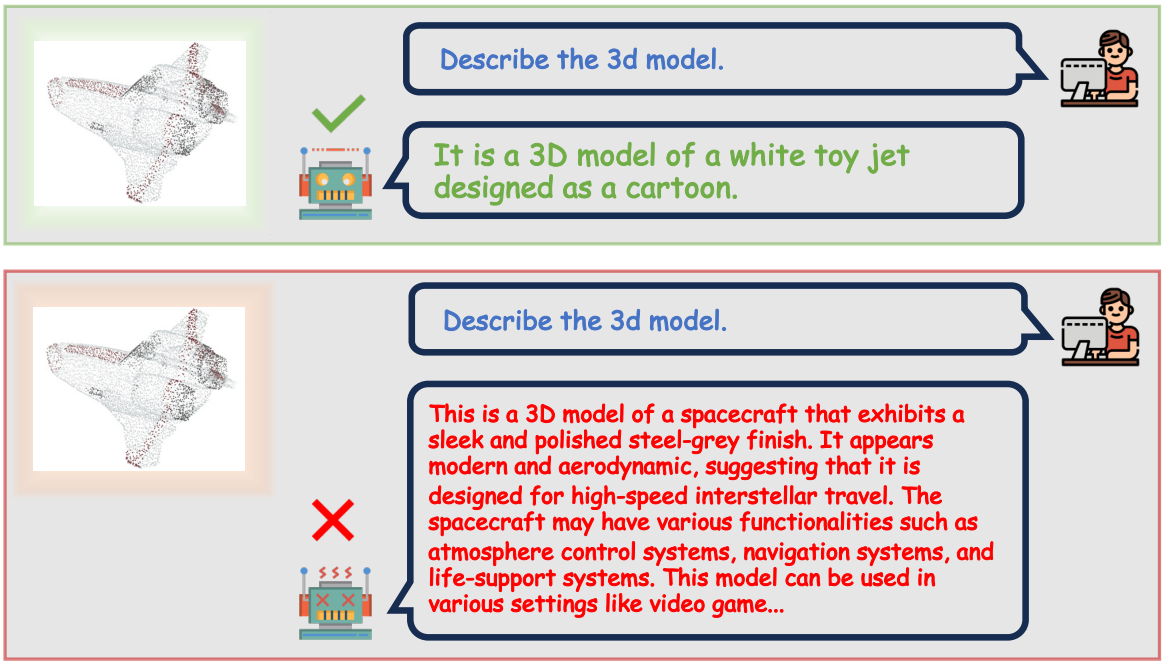


Figure 7: Example of Inflate3D attack on PointLLM with an Objaverse input.

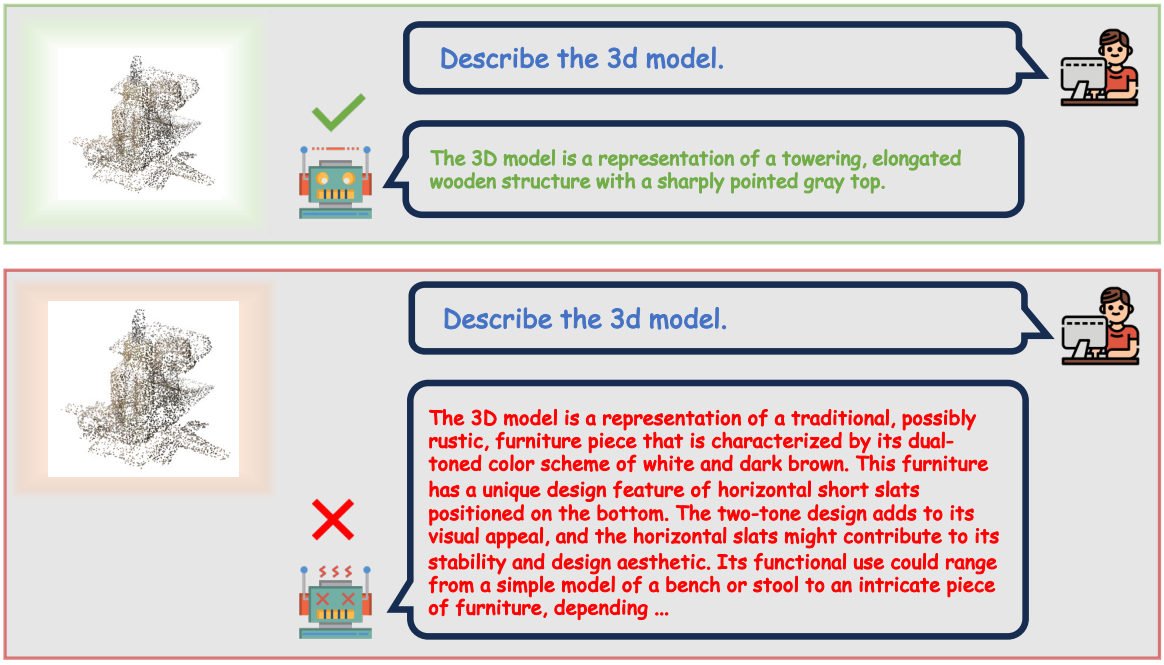


Figure 8: Example of Inflate3D attack on X-InstructBLIP with an Objaverse input.

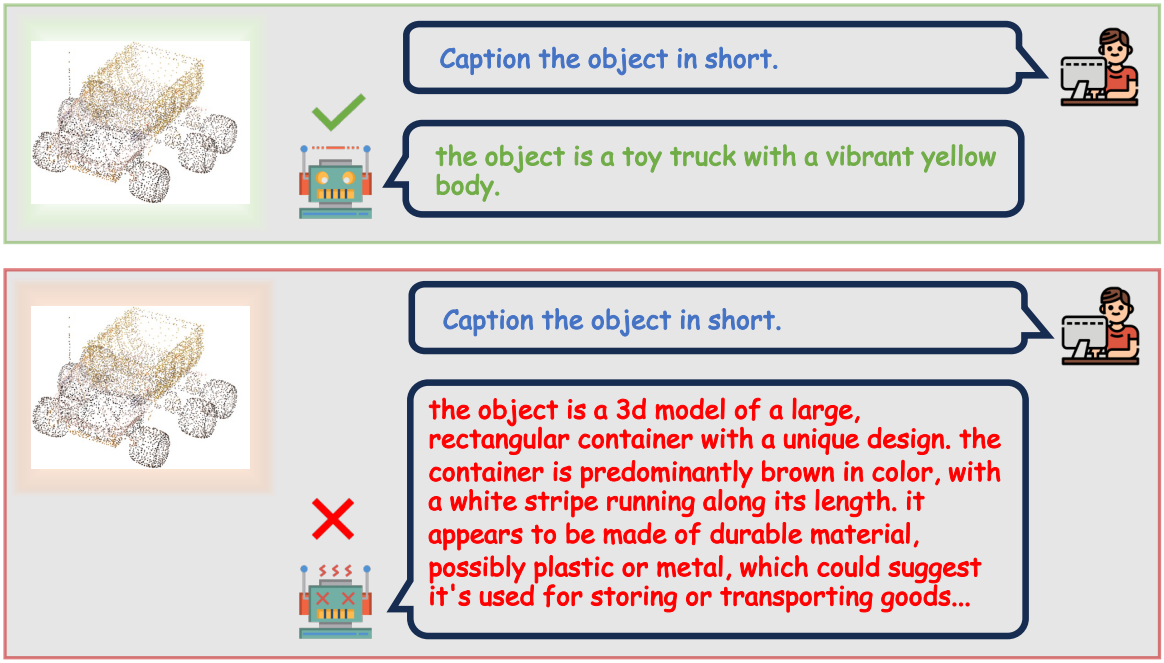


Figure 9: Example of Inflate3D attack on MiniGPT-3D with an Objaverse input.

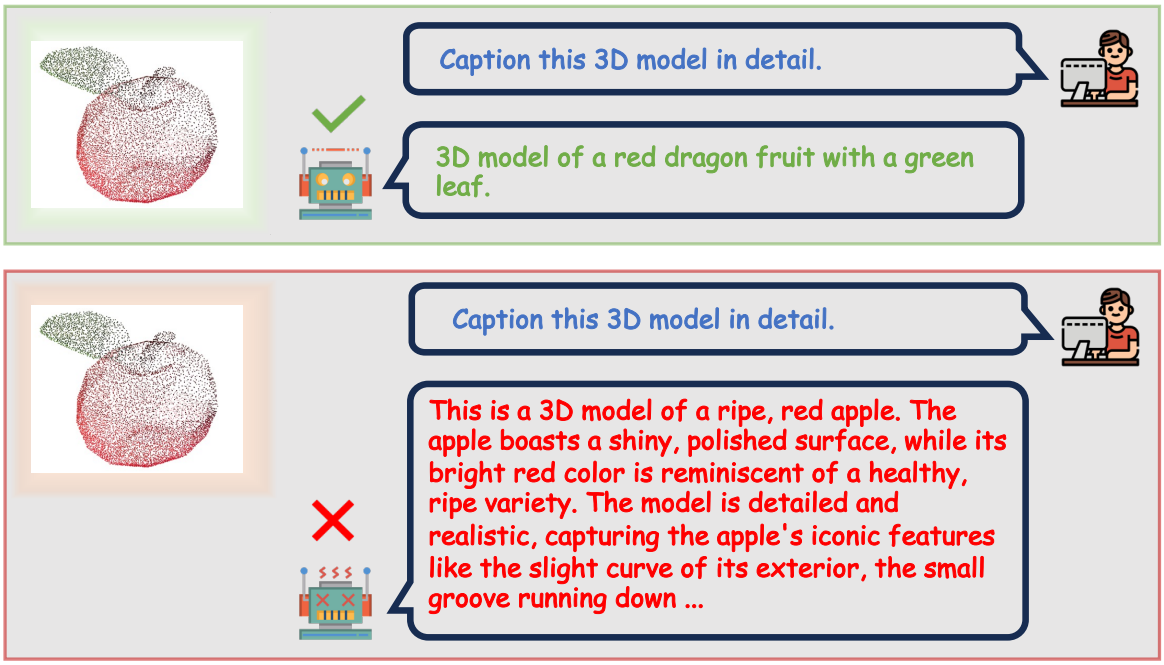


Figure 10: Example of Inflate3D attack on GreenPLM with an Objaverse input.

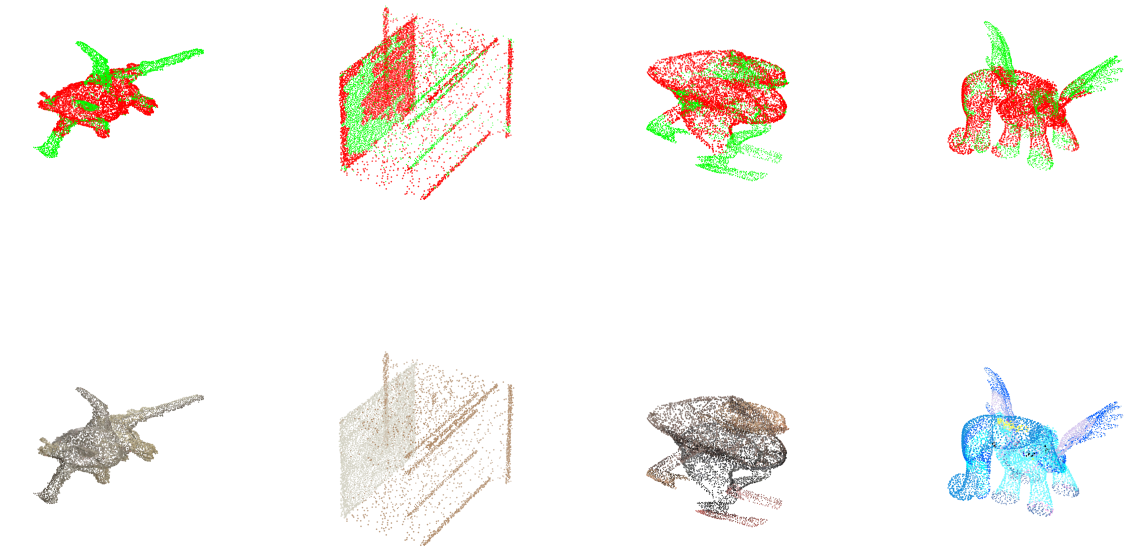


Figure 11: The visualization of our semantic-aware mask on Objaverse. The top row shows the masked object, while the bottom row shows the original object without mask. In the masked objects, points covered by the semantic-aware mask are highlighted in red, while the remaining unmasked points are shown in green.

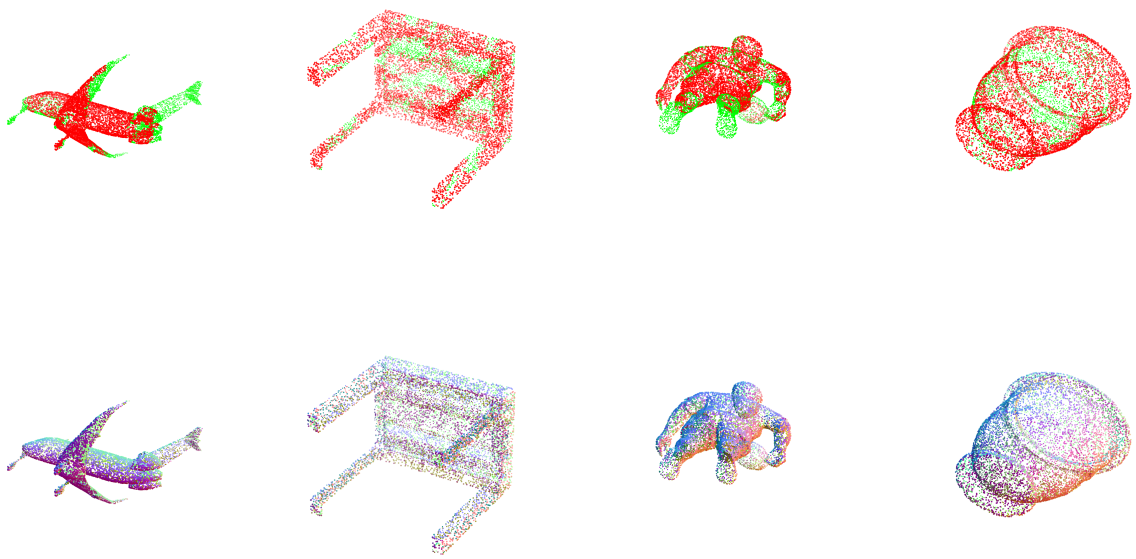


Figure 12: The visualization of our semantic-aware mask on ModelNet40. The top row shows the masked object, while the bottom row shows the original object without mask. In the masked objects, points covered by the semantic-aware mask are highlighted in red, while the remaining unmasked points are shown in green.