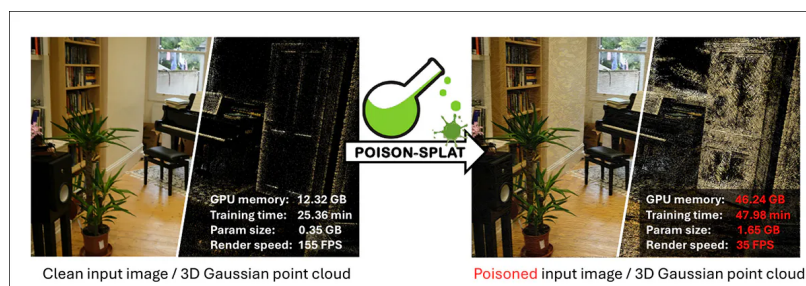


A Poisoning Attack Against 3D Gaussian Splatting

Originally published at <https://www.unite.ai/a-poisoning-attack-against-3d-gaussian-splatting/>



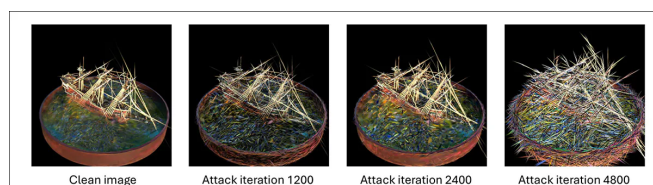
A new research collaboration between Singapore and China has proposed a method for attacking the popular synthesis method [3D Gaussian Splatting](#) (3DGS).



The new attack method uses crafted source data to overload the available GPU memory of the target system, and to make training so lengthy as to potentially inc equivalent to a denial-of-service (DOS) attack. Source: <https://arxiv.org/pdf/2410.08190>

The attack uses crafted training images of such complexity that they are likely to overwhelm an online service that allows users to create 3DGS representations.

This approach is facilitated by the adaptive nature of 3DGS, which is designed to add as much representational detail as the source images require for a realistic render. The method exploits both crafted image complexity (textures) and shape (geometry).



The attack system 'poison-splat' is aided by a proxy model that estimates and iterates the potential of source images to add complexity and Gaussian Splat instances to a model, until the host system is overwhelmed.

The paper asserts that online platforms – such as [LumaAI](#), [KIRI](#), [Spline](#) and [Polycam](#) – are increasingly offering 3DGS-as-a-service, and that the new attack method – titled *Poison-Splat* – is potentially capable of pushing the 3DGS algorithm towards 'its worst computation complexity' on such domains, and even facilitate a denial-of-service (DOS) attack.

According to the researchers, 3DGS could be radically more vulnerable other online neural training services. Conventional [machine learning](#) training procedures set parameters at the outset, and thereafter operate within constant and relatively consistent levels of resource usage and power consumption. Without the 'elasticity' that Gaussian Splat requires for assigning splat instances, such services are difficult to target in the same manner.

Furthermore, the authors note, service providers cannot defend against such an attack by limiting the complexity or density of the model, since this would cripple the effectiveness of the service under normal use.



From the new work, we see that a host system which limits the number of assigned Gaussian Splats cannot function normally, since the elasticity of these parameters is a fundamental feature of 3DGS.

The paper states:

'[3DGS] models trained under these defensive constraints perform much worse compared to those with unconstrained training, particularly in terms of detail reconstruction. This decline in quality occurs because 3DGS cannot automatically distinguish necessary fine details from poisoned textures.'

'Naively capping the number of Gaussians will directly lead to the failure of the model to reconstruct the 3D scene accurately, which violates the primary goal of the service provider. This study demonstrates more sophisticated defensive strategies are necessary to both protect the system and maintain the quality of 3D reconstructions under our attack.'

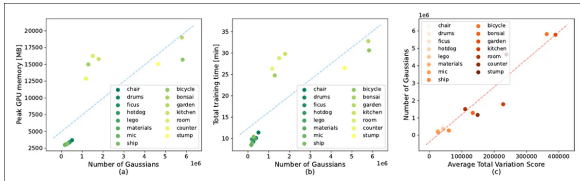
In tests, the attack has proved effective both in a loosely white-box scenario (where the attacker has knowledge of the victim's resources), and a black box approach (where the attacker has no such knowledge).

The authors believe that their work represents the first attack method against 3DGS, and warn that the neural synthesis security research sector is unprepared for this kind of approach.

The [new paper](#) is titled *Poison-splat: Computation Cost Attack on 3D Gaussian Splatting*, and comes from five authors at the National University of Singapore, and Skywork AI in Beijing.

Method

The authors analyzed the extent to which the number of Gaussian Splats (essentially, [three-dimensional ellipsoid 'pixels'](#)) assigned to a model under a 3DGS pipeline affects the computational costs of training and rendering the model.



The authors study reveals a clear correlation between the number of assigned Gaussians and training time costs, as well as GPU memory usage.

The right-most figure in the image above indicates the clear relationship between image sharpness and the number of Gaussians assigned. The sharper the image, the more detail is seen to be required to render the 3DGS model.

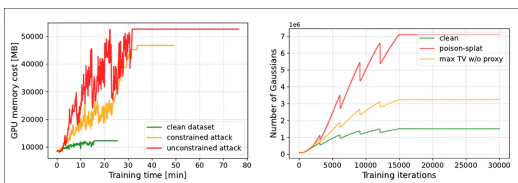
The paper states*:

'[We] find that 3DGS tends to assign more Gaussians to those objects with more complex structures and non-smooth textures, as quantified by the total variation score—a metric assessing [image sharpness](#). Intuitively, the less smooth the surface of 3D objects is, the more Gaussians the model needs to recover all the details from its 2D image projections.'

'Hence, non-smoothness can be a good descriptor of complexity of [Gaussians]'

However, naively sharpening images will tend to affect the semantic integrity of the 3DGS model so much that an attack would be obvious at the early stages.

Poisoning the data effectively requires a more sophisticated approach. The authors have adopted a *proxy model* method, wherein the attack images are optimized in an off-line 3DGS model developed and controlled by the attackers.



On the left, we see a graph representing the overall cost of computation time and GPU memory occupancy on the MIP-NeRF360 'room' dataset, demonstrating native performance, naive perturbation and proxy-driven data. On the right, we see that naïve perturbation of the source images (red) leads to quickly catastrophic results too early in the process. By contrast, we see that the proxy-guided source images maintain a more stealthy and cumulative attack method.

The authors state:

'It is evident that the proxy model can be guided from non-smoothness of 2D images to develop highly complex 3D shapes.

'Consequently, the poisoned data produced from the projection of this over-densified proxy model can produce more poisoned data, inducing more Gaussians to fit these poisoned data.'

The attack system is constrained by a 2013 Google /Facebook [collaboration](#) with various universities, so that the perturbations remain within bounds designed to allow the system to inflict damage without affecting the recreation of a 3DGS image, which would be an early signal of an incursion.

Data and Tests

The researchers tested poison-splat against three datasets: [NeRF-Synthetic](#); [Mip-NeRF360](#); and [Tanks-and-Temples](#).

They used the [official implementation](#) of 3DGS as a victim environment. For a black box approach, they used the [Scaffold-GS](#) framework.

The tests were carried out on a NVIDIA A800-SXM4-80G GPU.

For metrics, the number of Gaussian splats produced were the primary indicator, since the intention is to craft source images designed to maximize and exceed rational inference of the source data. The rendering speed of the target victim system was also considered.

The results of the initial tests are shown below:

Constrained Poison-splat attack with $\epsilon = 16/255$								
Metric	Number of Gaussians		Peak GPU memory [MB]		Training time [minutes]		Render speed [FPS]	
Scene	Setting	clean	poisoned	clean	poisoned	clean	poisoned	clean
NS-hodlog		0.185 M	1.147 M (6.20x)	3336	29747* (8.92x)	8.57	17.43 (2.03x)	443
NS-lego		0.341 M	0.805 M (2.36x)	3532	9960 (2.82x)	8.62	13.02 (1.51x)	349
NS-materials		0.169 M	0.410 M (2.43x)	3172	4886 (1.54x)	7.33	9.92 (1.35x)	447
NS-ship		0.272 M	1.071 M (3.94x)	3692	16666 (4.51x)	8.87	16.12 (1.82x)	326
TT-Courthouse		0.604 M	0.733 M (1.21x)	11402	12688 (1.11x)	13.81	14.49 (1.05x)	284
TT-Courtroom		2.890 M	5.450 M (1.89x)	9896	16388 (1.65x)	16.88	23.92 (1.42x)	139
TT-Museum		4.439 M	7.130 M (1.61x)	12704	18790 (1.48x)	19.79	26.38 (1.33x)	121
TT-Playground		2.309 M	4.307 M (1.87x)	8717	13032 (1.50x)	15.17	21.82 (1.44x)	151
MIP-bicycle		5.793 M	10.129 M (1.75x)	17748	27074* (1.53x)	33.42	44.44 (1.33x)	69
MIP-counter		1.195 M	4.628 M (3.87x)	10750	29796* (2.77x)	26.46	37.63 (1.42x)	164
MIP-room		1.513 M	7.129 M (4.71x)	12316	46238* (3.75x)	25.36	47.98 (1.89x)	154
MIP-stump		4.671 M	10.003 M (2.14x)	14135	25714* (1.82x)	27.06	38.48 (1.42x)	111

Unconstrained Poison-splat attack with $\epsilon = \infty$								
Metric	Number of Gaussians		Peak GPU memory [MB]		Training time [minutes]		Render speed [FPS]	
Scene	Setting	clean	poisoned	clean	poisoned	clean	poisoned	clean
NS-hodlog		0.185 M	4.272 M (23.09x)	3336	47859* (14.35x)	8.57	38.85 (4.53x)	443
NS-lego		0.341 M	4.159 M (12.20x)	3532	78852* (22.33x)	8.62	42.46 (4.93x)	349
NS-mic		0.205 M	3.940 M (19.22x)	3499	61835* (17.67x)	8.08	39.02 (4.83x)	300
NS-ship		0.272 M	4.317 M (15.87x)	3692	80956* (21.90x)	8.87	44.11 (4.97x)	326
TT-Courthouse		0.604 M	3.388 M (5.61x)	11402	29856* (2.62x)	13.81	25.33 (1.83x)	284
TT-Courtroom		2.890 M	13.196 M (4.57x)	9896	33871* (3.42x)	16.88	41.69 (2.47x)	139
TT-Museum		4.439 M	16.501 M (3.72x)	12704	43317* (3.41x)	19.79	48.89 (2.47x)	121
TT-Playground		2.309 M	10.306 M (4.46x)	8717	27384* (3.13x)	15.17	38.77 (2.56x)	151
MIP-bicycle		5.793 M	25.268 M (4.36x)	17748	63236* (3.56x)	33.42	81.48 (2.44x)	69
MIP-counter		1.195 M	11.167 M (9.34x)	10750	80732* (7.51x)	26.46	62.04 (2.34x)	164
MIP-room		1.513 M	16.019 M (10.59x)	12316	87540* (4.67x)	25.36	76.25 (3.01x)	154
MIP-stump		4.671 M	13.550 M (2.90x)	14135	36181* (2.56x)	27.06	51.51 (1.90x)	111

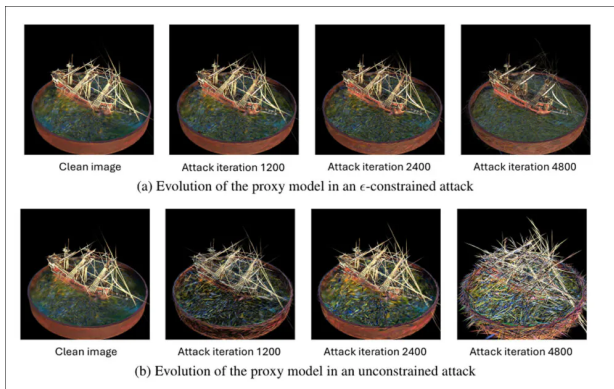
Full results of the test attacks across the three datasets. The authors observe that they have highlighted attacks that successfully consume more than 24GB of memory. Please refer to the source paper for better resolution.

Of these results, the authors comment:

'[Our] Poison-splat attack demonstrates the ability to craft a huge extra computational burden across multiple datasets. Even with perturbations constrained within a small range in [a constrained] attack, the peak GPU memory can be increased to over 2 times, making the overall maximum GPU occupancy higher than 24 GB.

[In] the real world, this may mean that our attack may require more allocable resources than common GPU stations can provide, e.g., RTX 3090, RTX 4090 and A5000. Furthermore [the] attack not only significantly increases the memory usage, but also greatly slows down training speed.

'This property would further strengthen the attack, since the overwhelming GPU occupancy will last longer than normal training may take, making the overall loss of computation power higher.'



The progress of the proxy model in both a constrained and an unconstrained attack scenario.

The tests against Scaffold-GS (the black box model) are shown below. The authors state that these results indicate that poison-splat generalizes well to such a different architecture (i.e., to the reference implementation).

Metric	Number of Gaussians		Peak GPU memory [MB]		Training time [minutes]	
Setting	clean	poisoned	clean	poisoned	clean	poisoned
Scene						
Constrained Black-box Poison-splat attack with $\epsilon = 16/255$ against Scaffold-GS						
NS-lego	0.414 M	2.074 M (5.01x↑)	3003	4808 (1.60x↑)	9.77	17.91 (1.83x↑)
NS-ship	1.000 M	3.291 M (3.29x↑)	3492	5024 (1.44x↑)	11.68	21.92 (1.88x↑)
MIP-bonsai	4.368 M	10.608 M (2.43x↑)	10080	12218 (1.21x↑)	35.33	37.71 (1.07x↑)
MIP-stump	6.798 M	14.544 M (2.14x↑)	7322	9432 (1.29x↑)	33.53	36.32 (1.08x↑)
Unconstrained Black-box Poison-splat attack with $\epsilon = \infty$ against Scaffold-GS						
NS-lego	0.414 M	3.973 M (9.60x↑)	3003	6242 (2.08x↑)	9.77	26.11 (2.67x↑)
NS-ship	1.000 M	4.717 M (4.72x↑)	3492	6802 (1.95x↑)	11.68	28.22 (2.42x↑)
MIP-bonsai	4.368 M	28.042 M (6.42x↑)	10080	22115 (2.19x↑)	35.33	78.36 (2.22x↑)
MIP-stump	6.798 M	34.027 M (5.01x↑)	7322	20797 (2.84x↑)	33.53	79.64 (2.38x↑)

Test results for black box attacks on NeRF-Synthetic and the MIP-NeRF360 datasets.

The authors note that there have been very few studies centering on this kind of resource-targeting attacks at inference processes. The 2020 paper [Energy-Latency Attacks on Neural Networks](#) was able to identify data examples that trigger excessive neuron activations, leading to debilitating consumption of energy and to poor latency.

Inference-time attacks were studied further in subsequent works such as [Slowdown attacks on adaptive multi-exit neural network inference](#), [Towards Efficiency Backdoor Injection](#), and, for language models and vision-language models (VLMs), in [NICGSlowDown](#), and [Verbose Images](#).

Conclusion

The Poison-splat attack developed by the researchers exploits a fundamental vulnerability in Gaussian Splatting – the fact that it assigns complexity and density of Gaussians according to the material that it is given to train on.

The 2024 paper [F-3DGS: Factorized Coordinates and Representations for 3D Gaussian Splatting](#) has already observed that Gaussian Splatting's arbitrary assignment of splats is an inefficient method, that frequently also produces redundant instances:

'[This] inefficiency stems from the inherent inability of 3DGS to utilize structural patterns or redundancies. We observed that 3DGS produces an unnecessarily large number of Gaussians even for representing simple geometric structures, such as flat surfaces.'

'Moreover, nearby Gaussians sometimes exhibit similar attributes, suggesting the potential for enhancing efficiency by removing the redundant representations.'

Since constraining Gaussian generation undermines quality of reproduction in non-attack scenarios, the growing number of online providers that offer 3DGS from user-uploaded data may need to study the characteristics of source imagery in order to determine signatures that indicate a malicious intention.'

In any case, the authors of the new work conclude that more sophisticated defense methods will be necessary for online services in the face of the kind of attack that they have formulated.

* My conversion of the authors' inline citations to hyperlinks

First published Friday, October 11, 2024

Writer on machine learning, domain specialist in human image synthesis. Former head of research content at Metaphysic.ai, until its dissolution into DNEG's Brahma.ai.

Portfolio site: martinanderson.ai

Contact: martin@martinanderson.ai



Discover More