EXPLOITING THE LAW OF PRÄGNANZ IN DIFFUSION MODELS TO PREVENT UNAUTHORIZED 3D STRUCTURE RECONSTRUCTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion-based 3D reconstruction models enable accurate generation of high-fidelity 3D structures with complex geometric details based on 2D visual signals. While these models facilitate powerful design and creation applications, they also introduce significant AI safety risks by enabling adversaries to infer sensitive 3D structures, such as proprietary 3D designs or confidential physical objects.

To address this threat, we present *Flatland*, a defense framework that protects structural information conveyed in 2D visual signals from being reconstructed by unauthorized 3D diffusion models. The key insight behind *Flatland* is an empirically observed simplicity bias in diffusion models: similar to human perception, their spatial reasoning behavior exhibits characteristics consistent with the psychological Law of Prägnanz, favoring simple structures over more complex alternatives. Leveraging this observation, *Flatland* injects simple structures into visual signals as low-amplitude perturbations, thereby disrupting unauthorized 3D reconstruction by inducing diffusion models to reconstruct the injected simple structures instead of the original sensitive ones. Experimental results demonstrate that *Flatland* effectively prevents 10 public diffusion models from reconstructing sensitive structures across 11 datasets. Moreover, we validate *Flatland* on 11 proprietary models from 9 commercial AI creation platforms, and analyze its robustness against perturbation removal and purification methods.

1 INTRODUCTION

Recent advances in scaling diffusion-based deep neural networks (DNN) have equipped artificial intelligence (AI) with spatial perception capabilities (Wu et al., 2025; Xiang et al., 2025; Hunyuan3D Team, 2025b; Chen et al., 2025b; Li et al., 2025a). These 3D diffusion models can perceive structural information from 2D visual signals and reconstruct the corresponding 3D structures with numerous digitized geometric primitives—vertices, edges, and faces. Diffusion-based 3D reconstruction enables diverse applications: artists utilize the models to generate 3D structures for game rendering (Fortune Business, 2026a) and animation production (Global Market Insights, 2026), while manufacturers integrate the models into supply chains to streamline the design of complex structural components (Fortune Business, 2026b). By substantially reducing the labor-intensive costs of manual 3D creation, diffusion-based 3D generative AI is catalyzing a market estimated to reach USD 51.8 billion (MarketsAndMarkets, 2026).

However, such powerful diffusion-based 3D reconstruction also introduce new AI safety risks: images may contain sufficient structural cues for diffusion models to recover sensitive 3D information, lowering the cost of copyright infringement. That is, adversaries can readily reconstruct proprietary 3D structures from publicly accessible high-resolution 2D images, enabling them to fabricate physical replicas using 3D printers (Horvath, 2026), such as creating counterfeit toys based on Disney's 2D cartoons, or the reproduction of marble sculptures from photographs taken in art galleries.

To address these issues, this study aims to protect sensitive structural information conveyed in visual signals from being reconstructed by unauthorized diffusion models. However, as summarized Table 2, the research problem remains largely unexplored; existing protection methods mainly focus on 2D visual semantics (Shan et al., 2024; Zhu et al., 2024; Zhou et al., 2025), such as protecting

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.

art styles (Shan et al., 2023), without considering 3D structural information, and perturbations that protect 2D semantics do not necessarily prevent structural reconstruction. Meanwhile, most existing methods require prior knowledge of known models (i.e., white-box settings (Le et al., 2023; Xue et al., 2024; Salman et al., 2023)), making it impossible to leverage their design principles to defend against the broad spectrum of newly developed models and proprietary black-box 3D diffusion models (Hunyuan3D Team, 2025d;c). Furthermore, these solutions incur prohibitive overhead due to interactive model queries (Zhou et al., 2025; Liu et al., 2024; Liang et al., 2023), making their design paradigm inapplicable for iteratively querying compute-intensive 3D diffusion models.

Our solution is inspired by methodologies from cognitive psychology (Van der Helm, 2000; Wannig et al., 2011). Surprisingly, we find that, like human cognition, diffusion models' spatial perception significantly conforms to the Law of Prägnanz—preferentially perceiving simple structural information (Wong, 2010a;b). That is, when simple and complex structures coexist in visual signals, the simple structures are prioritized for reconstruction even when only weakly represented, while the complex structures are suppressed. Such observations imply that simple structural signals act as universal interferences against unauthorized 3D diffusion models.

Based on the observations, this paper develops *Flatland* that injects *protective perturbation* to flat 2D images, such that diffusion models fail to infer sensitive structural information in the 2D visual signals for reconstructing structured 3D objects (see examples in Figure 1). Particularly, *Flatland* exploits the presence of the Law of Prägnanz effect in the diffusion models: it injects perturbations representing simple regular structures as *structural decoys* to trigger the simplicity bias, and thus induces the diffusion models to reconstruct the injected decoy structures instead of the sensitive ones. Specifically, *Flatland* injects low-amplitude perturbations that steer geometries in the visual signals toward regular objects with simple structures (e.g., glass cylinders) in a 2D feature space, thereby preventing 3D diffusion models from faithfully inferring the original structural information.

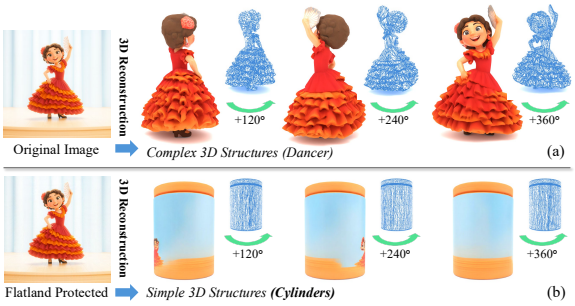


Figure 1: *Flatland* prevents unauthorized 3D diffusion reconstruction. By injecting protective perturbation, *Flatland* steers 3D diffusion models toward reconstructing an injected simple structure (cylinder) instead of the sensitive original (dancer).

Experimental results demonstrate that *Flatland* effectively defends against 10 state-of-the-art public 3D reconstruction models, preventing them from reconstructing sensitive structures from 11 datasets. That is, *Flatland* prevents all the models from producing structures visually similar to those in the datasets, where the quality of the produced structures significantly degrades, e.g., 44.79% F-Score decrease and 195.01% Chamfer Distance increase. Meanwhile, it incurs minimal quality degradation on protected visual signals. Moreover, *Flatland* prevents 11 proprietary AI models on 9 commercial AI creation platforms from inferring sensitive structural information. Furthermore, *Flatland* exhibits robustness against rescaling, recapturing, traditional noise filters (Hönig et al., 2025), and noise purification methods (Nie et al., 2022; Zhao et al., 2024).

In summary, the contributions of this paper are four-fold:

- We protect the structural information conveyed in 2D visual signals from being reconstructed into concrete 3D structures by diffusion models.
- Our study reveals that 3D diffusion models bias toward perceiving simple structures, which can serve as a universal interference against unauthorized reconstruction.
- We design *Flatland* protection system that exploits this simplicity bias by injecting simple structures as low-amplitude perturbations, misleading 3D diffusion models into reconstructing the injected decoy structures.
- We prototype *Flatland*[1] and validate its accuracy, efficiency, and robustness across various public and proprietary 3D diffusion models on different datasets.

[1]Source code: https://anonymous.4open.science/r/Flatland-7777.

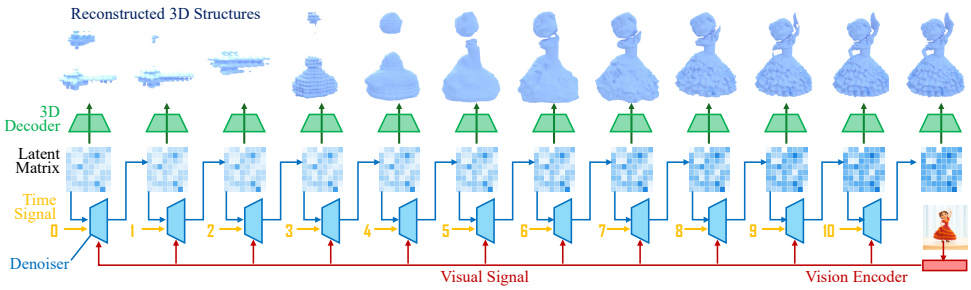


Figure 2: Reconstruction process of 3D diffusion model: from constructing general structures to details.

2 Problem Statement

Threat Model. We consider that an adversary seeks to derive high-quality 3D structures using diffusion models based on 2D visual contents published by a victim in the format of images with RGB channels, such as franchised 2D cartoon characters. The adversary may execute arbitrary public or proprietary models, without revealing any model-related information, including the architectures, parameters, and training datasets. Particularly, the adversary knows the victim's images may be protected by *Flatland*. Therefore, the adversary tries to remove the protective perturbation, for example, by applying recent diffusion-based purification (Nie et al., 2022; Zhao et al., 2024), as well as traditional methods, e.g., Gaussian filtering and JPEG compression. Furthermore, we assume the defense pipeline of *Flatland* is transparent, i.e., its detailed algorithm and parameters are publicly known. This enables the adversary to obtain sample pairs of images before and after adding *Flatland*'s perturbation, allowing them to train deep-learning-based noise removal models (e.g., LightShed (Foerster et al., 2025)). Additionally, the adversary may also randomly rescale and crop the images in an attempt to invalidate *Flatland*.

Design Goals. To protect the 2D images from being reconstructed as 3D objects, *Flatland* adds protective perturbations to the victim's images, which incur minimal degradation in image quality but disrupt the 3D reconstruction. *Flatland* should achieve the following goals:

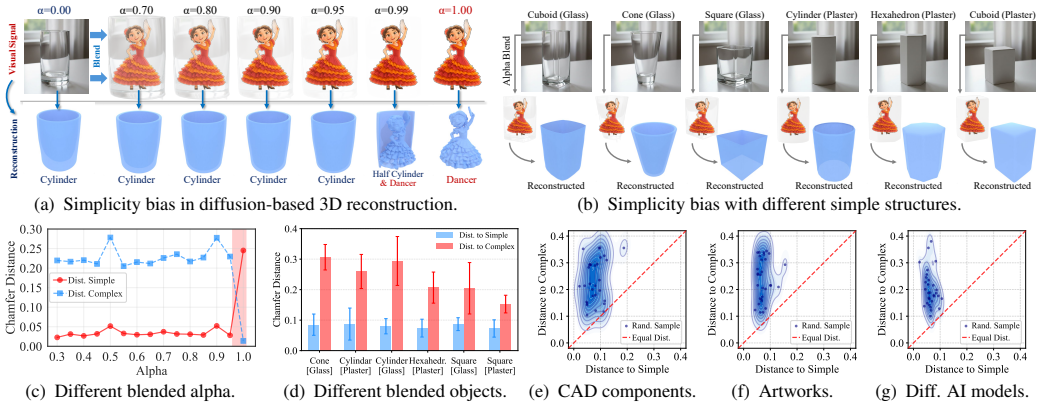
1. **Generic Protection:** *Flatland* should not rely on any prior knowledge of the models, such as architectures, parameters, and training datasets, thereby maintaining effectiveness across a broad spectrum of public or proprietary fast-evolving 3D AI models (Hunyuan3D Team, 2025c).
2. **Non-Interactive Protection:** *Flatland* should avoid querying the models, because repetitively querying the 3D generation models incurs prohibitively high latency and overhead, which can be easily detected (Li et al., 2022). Moreover, *Flatland* cannot build or query surrogate models similar to the adversary's models, due to the overhead of constructing and querying such surrogates.
3. **Robust Protection:** *Flatland* should be resilient to image editing operations (e.g., rescaling and recapturing), noise filters (e.g., Gaussian filters), especially diffusion-based noise purification methods (Nie et al., 2022), and adversarially trained noise removers (Foerster et al., 2025).

In general, this study assumes a fully model-agnostic scenario with no prior knowledge or query access. In this setting, *Flatland* must effectively disrupt unauthorized 3D reconstruction, preserve image quality, and maintain robustness against perturbation removal behaviors.

3 Key Observation

3.1 Process of Diffusion-Based 3D Reconstruction

The diffusion-based 3D reconstruction pipeline typically comprises three components: a vision encoder, a denoiser, and a 3D decoder (Song et al., 2021). The architecture is illustrated in Figure 2. The process begins with a randomized latent matrix serving as the initial internal representation. At each step, the denoiser refines the latent matrix from the preceding step based on two inputs: (i) the temporal embedding of the time-step and (ii) the visual features extracted by the vision encoder (Oquab et al., 2024). Finally, the decoder translates the refined latent representation into a concrete 3D output, consisting of a dense collection of faces, edges, and vertices. As shown, the diffusion model gradually enrich the 3D structure with details by iteratively denoising the latent.



(a) Simplicity bias in diffusion-based 3D reconstruction. (b) Simplicity bias with different simple structures.

(c) Different blended alpha. (d) Different blended objects. (e) CAD components. (f) Artworks. (g) Diff. AI models.

Figure 3: Diffusion-Based 3D Reconstruction Exhibits Simplicity Bias. (a) When simple and complex structures coexist, the model reconstructs the simple ones. (b) Regardless of shape or material, simple structures blended at $\alpha = 10\%$ still dominate the reconstruction. (c) As α varies, outputs stay close to the simple objects and far from the complex ones. (d) The same pattern holds across shapes and materials. (e)-(g) On CAD components (e), artworks (f), and across models (g), the distance to simple objects is consistently lower.

3.2 Simplicity Bias in 3D Diffusion Models

The Law of Prägnanz (Koffka, 2013; Wagemans et al., 2012a;b) is a fundamental principle in psychology (Wong, 2010a;b), asserting that human perceptual processes inherently gravitate toward simplicity (Van der Helm, 2000; Feldman, 2003). For instance, when observing a dancer on stage, human perception first recognizes the gestural posture—such as the simple spatial relationships of the head and limbs—before recognizing the intricate structures of ornamentation on the attire. In general, human cognition is *lazy* (Van der Helm, 2000; Wannig et al., 2011), automatically prioritizing the perception of simple geometries while suppressing complex components.

We first observe that AI models' spatial perception significantly conforms to the Law of Prägnanz: when simple and complex structural information coexist, they exhibit a strong bias toward reconstructing simple structures. In practice, we cropped and aligned two images—one depicting a cylinder glass with simple regular structures and another depicting a complex dancer—then superimposed them via alpha blending as the visual signal input. Figure 3(a) shows that even when the simple structure accounts for merely 5% of the alpha blending, making it nearly invisible to the human visual system, the 3D diffusion model consistently prioritizes reconstructing the simple structure.

We further verify this observation through a quantitative analysis. Specifically, we measure the distance between the reconstructed structures and ground-truth simple/complex structures. Figure 3(c) shows that, when the ratio of blended simple objects ranges between $0.30 \sim 0.95$ alpha, the reconstructed structures are consistently similar and close to the simple objects. The distance to the simple objects is $5.329 \sim 9.673$ times lower than the distance to the complex ones. In Figure 3(b), we also blend different objects with various materials by fixing alpha to 10%. In this case as well, the reconstructed structures are similar to the simple objects, where the associated distances are $27.78\% \sim 47.55\%$ of the distance to the complex objects (see Figure 3(d)). Such observations also hold for different datasets and AI models (Hunyuan3D Team, 2025b; Xiang et al., 2025), according to the distance distribution in Figures 3(e), 3(f), and 3(g). For comparison, we blend 10^2 randomly selected complex objects from the Hunyuan3D-2.1 curated dataset at $\alpha = 0.9$: the reconstructions do not change noticeably (≤ 0.05 CD variation); injecting the same-energy random signals, which simulate unseen OOD structures, is also ineffective. In conclusion, the diffusion models exhibit *lazy* spatial perception, preferentially perceiving simple structural information in their presence. We further confirm the conclusion by identifying the internal mechanism behind this bias through neuron-activation pattern analysis: visual signals conveying simple structures suppress neuron activation in diffusion models. Due to space limits, we defer the details to Appendix A; these results imply that simple-structure signals act as effective interference against the diffusion processes.

4 Design of *Flatland*

Based on the observations, we exploit the bias on simple 3D structures to develop *Flatland*. Our method injects signals of simple structures into the 2D visual input. The 3D diffusers then prior-

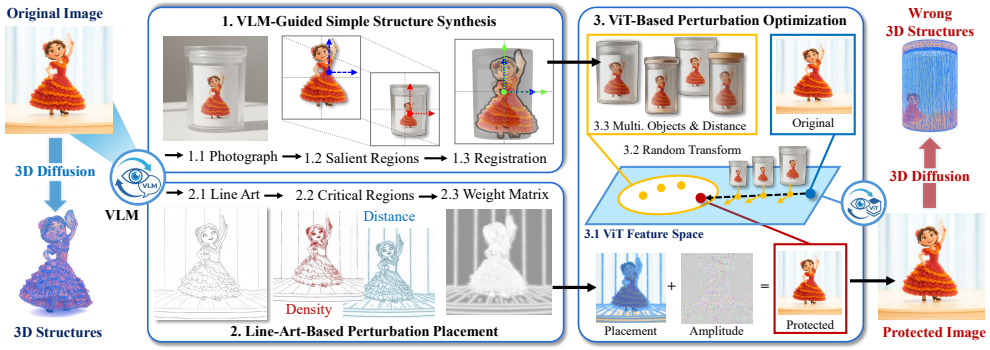


Figure 4: Architecture of *Flatland* sensitive 3D structure protection system.

itize reconstructing the injected simple structure, consequently failing to faithfully reconstruct the sensitive structures depicted on the protected images. However, *Flatland* cannot simply blend simple objects as in Section 3, because alpha blending degrades visual quality and makes the injected structures perceptible to human observers. In practice, *Flatland* adds perturbation to inject simple structural information by steering the original image toward objects with simple structures in a visual feature space. We mainly use regular cylinders as the simple structures, because, as Figure 3(d) shows, cylindrical structures effectively interfere with the reconstruction process. Eventually, the models are misled into reconstructing these simple cylindrical structures rather than the sensitive structures. Overall, *Flatland* includes three key modules, as shown in Figure 4. Due to the page limit, we provide the formal specifications of *Flatland*'s design in Appendix B.

VLM-Guided Simple Structure Synthesis. This module synthesizes images depicting simple objects—the structures to be injected by the perturbation. We instruct a Vision-Language Model (VLM) to paint the original image on flat paper, place the paper within a translucent cylindrical container, and synthesize a photograph with rich lighting and shading effects. The synthesized photograph closely resembles the original one, which allows low-amplitude perturbation to inject the simple structures by minimally moving the original image toward the synthesized one. Finally, the module identifies salient regions in the photograph and aligns the regions with the original image.

Line-Art-Based Perturbation Placement. This module allocates higher-amplitude perturbations to critical regions of the original image while reducing perturbations in other areas, thereby minimizing visual quality degradation. Initially, it generates a line art based on the original images, to identify regions with critical structural information, as line art tends to highlight significant structural changes. Subsequently, it applies density- and distance-based analysis to assign importance scores to regions surrounded by these lines. Finally, based on the scores, the module smooths region boundaries and computes a weight matrix that serves as the upper bound for perturbation amplitudes.

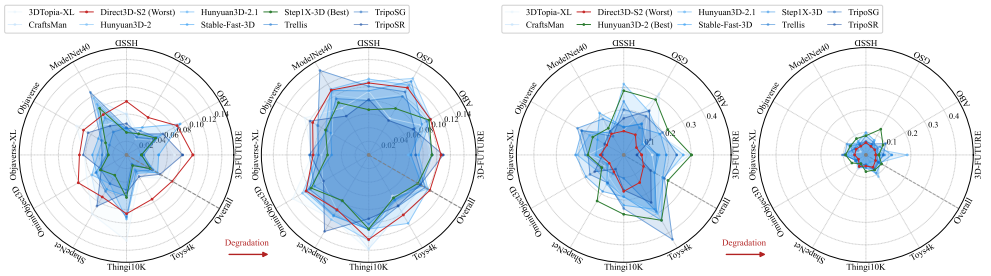
ViT Feature Space Perturbation Optimization. This module determines the optimal amplitude subject to the upper bound. To balance tractability and effectiveness, we conduct the optimization within the condensed feature space of a Vision Transformer (ViT). Specifically, the module jointly optimizes perturbations across multiple targets, i.e., it simultaneously gravitates the original image toward multiple simple structures synthesized by the VLM. This allows the injected perturbations to effectively represent simple structures and mislead AI models. To ensure robustness, during the optimization, we use different distance metrics and random transformations to eliminate distinctive patterns that could be exploited by perturbation removal techniques.

5 Experiments

The protected sensitive structures cover 11 public datasets across four categories: scanned objects, CAD components, community artworks, and traditional structure datasets. In addition, we evaluate on curated datasets provided by the ten public models. For the quantitative analysis, we mainly use the increase of Chamfer Distance (CD) and decrease of F-Score as the metrics of reconstruction quality degradation. We also employ six other metrics spanning three categories: point-cloud-based, feature-space-based, and spatial-based. Details of the implementation, targeted AI models, datasets, and definitions of the metrics can be found in Appendix D.

Table 1: Chamfer distance relative degradation (%) on public datasets (higher is better).

Methods	Scanned Objects		CAD Modeling				Community Artworks			Benchmark		Avg.
	GSO	Omini.3D	HSSD	3D-FUT.	ABO	Thingi10K	Obj.-XL	Objaverse	Toys4k	ModelNet	ShapeNet	
3DTopia-XL	103.46	92.85	47.45	36.30	51.60	87.99	43.10	48.50	84.58	77.12	87.62	69.14
CraftsMan	528.15	32.21	76.09	436.91	166.46	419.59	143.28	153.85	396.70	63.35	134.54	231.92
Direct3D-S2	98.20	39.35	39.30	46.78	48.72	87.93	38.46	34.55	122.89	88.60	60.49	64.12
Hunyuan3D-2	836.69	156.31	465.68	276.55	265.29	450.99	164.43	195.00	492.24	97.98	466.59	351.61
Hunyuan3D-2.1	390.19	143.01	454.23	367.79	227.09	283.43	134.03	170.84	576.52	112.30	247.55	282.45
Stable-Fast-3D	121.77	98.55	73.34	222.84	90.24	369.92	47.32	60.53	266.35	163.40	53.35	142.51
Step1X-3D	384.74	335.84	248.95	201.84	139.55	290.55	138.57	163.69	347.31	98.65	323.58	243.02
Trellis	258.29	87.01	188.39	294.23	145.19	69.89	251.20	188.53	383.69	251.16	52.02	197.24
TripoSG	426.08	220.06	205.44	264.42	138.20	389.33	153.24	223.91	502.11	90.96	214.28	257.09
TripoSR	228.01	154.00	107.33	48.34	21.38	135.62	44.05	58.08	344.45	61.96	17.81	111.00
Average	337.56	135.92	190.62	219.60	129.37	258.52	115.77	129.75	351.68	110.55	165.78	195.01



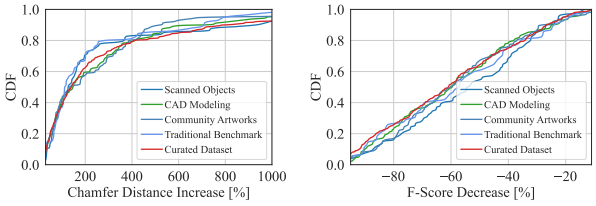
(a) Chamfer distance degradation. (b) F-Score degradation.

Figure 5: Absolute 3D reconstruction quality degradation indicates the effectiveness of *Flatland* protection.

5.1 Quantitative Evaluation

Performance on Public Datasets. We first analyze the relative degradation introduced by *Flatland*, with detailed per-method CD results in Table 1 (see distribution in Figure 6, and F-Score results in Table 5). We observe that *Flatland* significantly degrades the performance of unauthorized 3D reconstruction, i.e., it introduces a 195.01% average CD increase and a 44.79% average F-Score decrease. Overall, these 3D models suffer at least a 64.12% CD increase, indicating that *Flatland*'s protection is effective across different model architectural designs. Moreover, the relative degradation varies across dataset categories: scanned objects show 236.74% average CD degradation, CAD components 202.03%, and community artworks 198.90%. We further conduct cross-metric analysis in Table 6 in Appendix E.

Beyond these relative metrics, Figure 5 reports the absolute reconstruction quality before and after protection. The radar charts demonstrate substantial degradation: CD increases from 0.055 to 0.096 on average, with individual models ranging from 0.041 $\sim$ 0.077 originally and degrading to 0.083 $\sim$ 0.104 after protection. Similarly, F-Score decreases from 0.175 to 0.065, with original values spanning 0.102 $\sim$ 0.228 dropping to 0.050 $\sim$ 0.081. For instance, the reconstruction quality of Hunyuan3D-2 drops significantly, with CD rising from 0.041 to 0.100 (+143.90%) and F-Score decreasing from 0.228 to 0.068 ($-$70.17%). Results on curated datasets provided by the model designers are deferred to Appendix E, where *Flatland* still delivers 224.27% average CD degradation and 51.17% F-Score degradation.



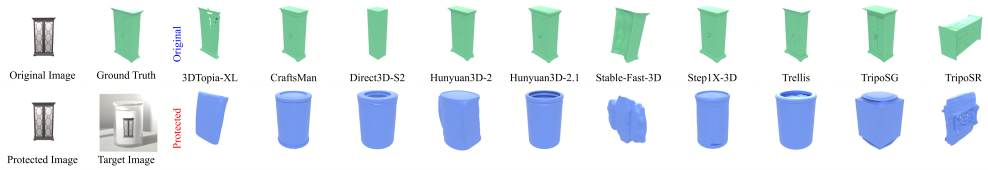
(a) Chamfer Distance distribution. (b) F-Score distribution.

Figure 6: Distribution of *Flatland*'s protection effectiveness.

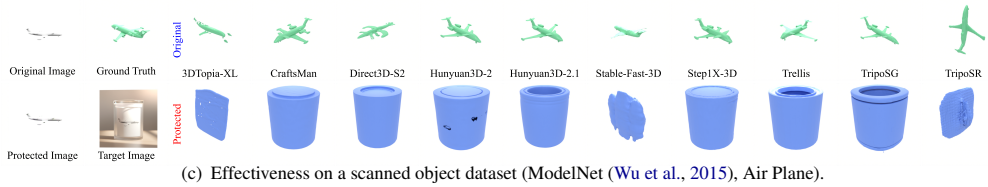
Comparison with Existing Protection. We compare *Flatland* against ten representative 2D image protection methods (details in Appendix C). No existing method can prevent diffusion-based unauthorized 3D reconstruction, achieving only -0.15% to 8.13% average CD degradation across all models. Specifically, Anti-DreamBooth (Le et al., 2023) (0.06%), Dormant (Zhou et al., 2025) (0.39%), and MetaCloak (Liu et al., 2024) (2.19%) show nearly no impact on 3D reconstruction quality. Even the best-performing baseline, AdvDM (Liang et al., 2023), achieves only 8.13% degradation—insufficient to prevent meaningful 3D reconstruction. Notably, some methods even yield negative degradation on certain models, such as Nightshade (Shan et al., 2024) on Direct3D-S2 (-4.50%) and SDS (Xue et al., 2024) on Direct3D-S2 (-9.50%), indicating that they may inadver-



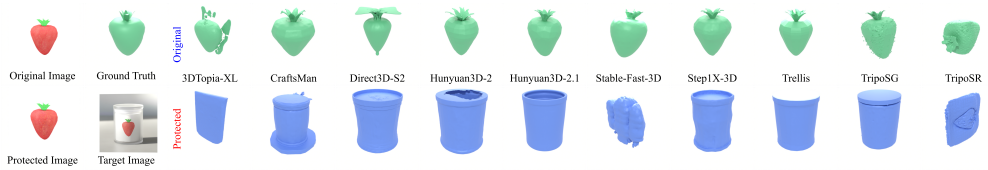
(a) Protection effectiveness for proprietary AI models.



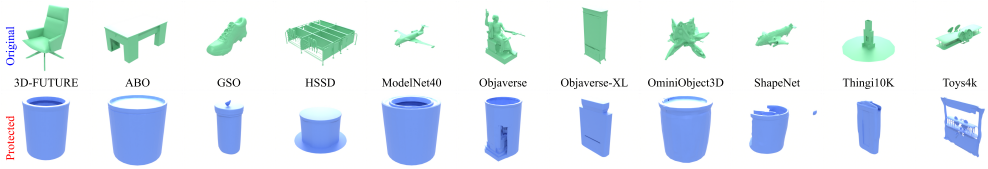
(b) Effectiveness on a CAD component dataset (3D-FUTURE (Fu et al., 2021), Closet).



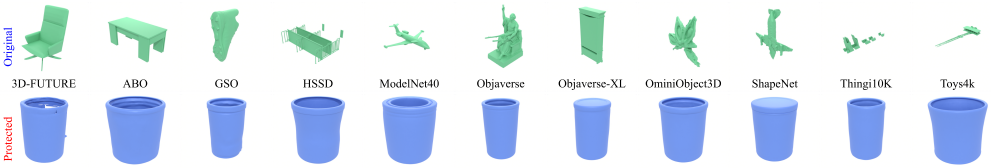
(c) Effectiveness on a scanned object dataset (ModelNet (Wu et al., 2015), Air Plane).



(d) Effectiveness on an artwork dataset (Objeverse-XL (Deitke et al., 2023b), Strawberry).



(e) Protection effectiveness for Trellis model (Xiang et al., 2025).



(f) Protection effectiveness for Hunyuan3D-2.1 model (Hunyuan3D Team, 2025b).

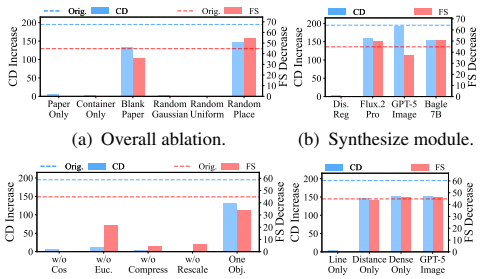
Figure 7: Visual analysis for protection effectiveness of *Flatland* against public and proprietary AI models.

tently improve 3D generation quality. In contrast, *Flatland* achieves 195.01% average degradation across all models, outperforming the best baseline by 178.36% on average. *Flatland* achieves at least 67.14% additional degradation on 3DTopia-XL, 181.52% on CraftsMan, 56.12% on Direct3D-S2, and up to 347.21% on Hunyuan3D-2. Additionally, Figure 8 presents ablation studies that quantify the contribution of each design component (details in Appendix G).

5.2 Qualitative Evaluation

We render reconstructed 3D structures with and without the protection of *Flatland*, as well as ground-truth structures. Due to space limits, we present one example for each dataset in the main text. These examples are representative of the best original reconstruction fidelity w.r.t. the Chamfer Distance to ground-truth structures (w/o protection). For dataset-wise comparison, we present the results of Hunyuan3D-2.1 (Hunyuan3D Team, 2025b) and Trellis (Xiang et al., 2025), because only these two models can consistently produce high-quality 3D structures. Additionally, we also validate that *Flatland* incurs minimal decrease in the quality of visual signals.

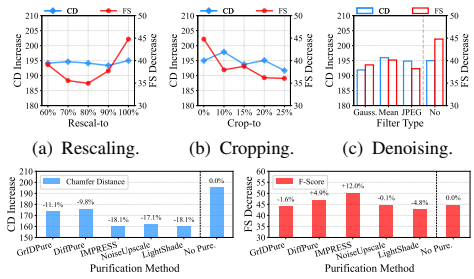
Protection Effectiveness on Commercial Platforms. We evaluate *Flatland* across nine AI-powered 3D creation platforms, comprising 11 proprietary AI models. These platforms allow users to upload 2D images, from which the AI models infer structural information and generate 3D struc-



(a) Overall ablation. (b) Synthesize module.

(c) Refinement module. (d) Placement module.

Figure 8: Ablation study by modules for *Flatland*.



(a) Rescaling. (b) Cropping. (c) Denoising.

(d) Purification (CD). (e) Purification (F-Score).

Figure 9: Adversarial behaviors against *Flatland*.

tures. As shown in Figure 7(a), when enabling *Flatland* for protection, the proprietary models fail to reconstruct meaningful structures based on images with *Flatland*'s protective perturbations. Consequently, they produce the simple structures injected by *Flatland*, rather than the intricate details of the original dancer. Notably, Tripo Studio fails to reconstruct the dancer and instead generates a square; this occurs because *Flatland* also injects the square structure of a painted paper to maintain visual similarity, as discussed in Section 4. Due to the high cost and limited usage quotas of these proprietary models, they are used primarily for qualitative evaluation.

Different Models. Figures 7(b), 7(c), and 7(d) show that *Flatland* protects structural information from being reconstructed by all the AI models (see triplane analysis in Figures 18 and 19 in Appendix F). We observe that AI models using different 3D representations all fail to reconstruct meaningful structures based on protected images, including voxel approaches (e.g., Direct3D-S2 (Wu et al., 2025)), Gaussian methods (e.g., Trellis (Xiang et al., 2025)), and TSDF representations (e.g., CraftsMan3D (Li et al., 2025a)). Moreover, *Flatland* is also effective for traditional models with triplane representations (e.g., Stable-Fast-3D (Boss et al., 2025)). Notably, some methods consistently reconstruct the square structure of the painted paper (e.g., 3DTopia-XL), while others produce the translucent cylinders outside the paper. This indicates that different architectural designs interpret the protected images in distinct ways, yet all fail to reconstruct the protected structures.

Different Datasets. Figures 7(e) and 7(f) demonstrate that *Flatland* effectively protects images across different datasets. More examples can be found in Figure 16 and Figure 17 in Appendix F. For scanned objects, which contain high-fidelity structures captured from real-world items, *Flatland* consistently prevents the models from reconstructing the protected structures. On CAD modeling datasets, which include precisely designed industrial products, the protected images consistently yield minimal structures instead of the detailed sensitive geometries. For community artworks, characterized by diverse artistic styles, *Flatland* successfully degrades the reconstruction quality to flat papers or cylindrical containers.

5.3 Robustness Evaluation

Rescaling. Figure 9(a) evaluates *Flatland*'s robustness against image rescaling by downsampling protected images to 60 $\sim$ 90% of their original resolution. *Flatland* maintains strong protection across all ratios, achieving 193.38 $\sim$ 194.65% CD degradation, representing only a 0.37 $\sim$ 0.87% decrease from the original performance (195.01%). For F-Score, *Flatland* achieves 34.95 $\sim$ 39.11% degradation, with at most a 9.84% decrease. Notably, aggressive rescaling (60 $\sim$ 70%) does not significantly weaken protection, as severe downsampling degrades overall image quality and inadvertently hinders 3D reconstruction—complementing rather than undermining *Flatland*'s protection.

Cropping. Figure 9(b) evaluates *Flatland*'s robustness against cropping-based recapturing by randomly removing 10 $\sim$ 25% of the image. *Flatland* maintains strong resilience across all ratios, achieving 191.70 $\sim$ 197.90% CD degradation, with only a 1.28 $\sim$ 3.31% decrease from the original performance. Notably, 10% cropping slightly improves CD degradation (+2.89%), as border removal and padding introduce additional interference that further disrupts 3D reconstruction.

Denoising. Figure 9(c) evaluates *Flatland*'s resilience against Gaussian filtering, mean filtering, and JPEG compression. *Flatland* achieves 191.89% (Gaussian), 195.98% (Mean), and 194.84% (JPEG) CD degradation, representing only a 0.87% $\sim$ 3.12% decrease from the original performance. For F-Score, *Flatland* achieves 38.21% $\sim$ 40.13% degradation. Notably, mean filtering slightly improves CD degradation (+0.97%), as denoising blurs image details required for accurate 3D reconstruction.

Purification. Figures 9(d) and 9(e) present CD and F-Score degradation under five state-of-the-art noise purification methods. Despite applying GrIDPure (Zhao et al., 2024), DiffPure (Nie et al., 2022), IMPRESS (Cao et al., 2023), and NoisyUpscale (Hönig et al., 2025), none of these diffusion-based methods can effectively decrease *Flatland*'s protection effectiveness, with CD degradation remaining in the range of 159.66 $\sim$ 175.86% and F-Score degradation maintained at 44.09 $\sim$ 50.17%. LightShed (Foerster et al., 2025) represents the strongest purification, as it trains autoencoders on protected image examples to learn and remove specific perturbation patterns. Under the purification of LightShed, *Flatland* achieves 159.74% CD degradation (-18.09%) and 42.62% F-Score degradation (-4.84%), maintaining substantial degradation across the models, such as 232.60% on Hunyuan3D-2.1, which is $19.6\times$ higher than the baselines' degradation in Section 5.1.

6 Related Work

Protection for 2D Diffusion. Existing protection methods perturb images to prevent diffusion models from exploiting 2D semantics for generation. As shown in Table 2, they target different stages of the pipeline. During pretraining, Glaze (Shan et al., 2023) prevents AI from learning artists' styles, Nightshade (Shan et al., 2024) corrupts the learning of concepts on images, and Mist (Liang & Wu, 2023) disrupts artwork recreation. During fine-tuning, MetaCloak (Liu et al., 2024) and Anti-DreamBooth (Le et al., 2023) prevent images personalization. At inference time, PhotoGuard (Salman et al., 2023), SDS (Xue et al., 2024), and VOID (Qiu et al., 2026) prevent image editing; Zhu *et al.* (Zhu et al., 2024) disable instructed drawing; AdvDM (Liang et al., 2023) prevents style transfer; Dormant (Zhou et al., 2025) blocks motion generation.

3D Reconstruction Models. Diffusion models enable 3D reconstruction (Zhang et al., 2024; Chen et al., 2025a; Xu et al., 2024; Zhao et al., 2023), which adopt unified latent vector sets (Zhang et al., 2023) or latents associated with voxels or triplanes (Xiang et al., 2025; Li et al., 2025c), as internal representation of 3D structures. Initially, 3DShape2VecSet (Zhang et al., 2023) proposed a global vector set to represent 3D space; CLAY (Zhang et al., 2024) integrated the vector set with diffusion models (Rombach et al., 2022); the Hunyuan3D (Hunyuan3D Team, 2025c;d;a;b) series adopted the flow-matching framework, Step1X-3D (Li et al., 2025b) and TripoSG (Li et al., 2025c) leveraged the rectified-flow paradigm, Dora (Chen et al., 2025a) improved sampling for vector construction. Another line of work assigns one latent per spatial cell, e.g., voxel-based latents in Direct3D-S2 (Wu et al., 2025) and flexible regional latents in 3DTopia-XL (Chen et al., 2025b); such latents can also encode color and material properties (Wu et al., 2025). Trellis (Xiang et al., 2025) and XCube (Ren et al., 2024) further enhance the sparsity of the voxel-based latent. In addition, autoregressive approaches treat triplanes as a sequence of latent tokens (Boss et al., 2025; Li et al., 2025a).

Noise Purification. DiffPure (Nie et al., 2022) leverages diffusion models' denoising ability to purify adversarial noise for image classification, further improved by GrIDPure (Zhao et al., 2024) through patch-wise processing. Recent studies focus on purifying protective perturbations. IMPRESS (Cao et al., 2023) reveals that Glaze (Shan et al., 2023) and PhotoGuard (Salman et al., 2023) protections can be removed using diffusion models. Hönig *et al.* (Hönig et al., 2025) observed low robustness of existing protections for diffusion-based style transfer, and developed simple removal strategies. LightShed (Foerster et al., 2025) considered that adversaries may obtain protected examples and train autoencoders to remove perturbations.

7 Conclusion

This paper presents *Flatland*, a protection system that prevents unauthorized 3D diffusion models from reconstructing sensitive structural information conveyed in 2D visual signals. By revealing that 3D diffusion models conform to the psychological Law of Prägnanz and exhibit a fundamental preference for reconstructing simple structures, *Flatland* injects simple structures into visual signals as low-amplitude perturbations to mislead diffusion models into reconstructing the injected structures instead of the sensitive ones. Experiments across 11 proprietary and 10 public diffusion models on 11 datasets confirm that *Flatland* effectively protects sensitive structures, illustrating the practicality of leveraging psychological principles to mitigate AI safety risks. We discuss the current limitations and future directions of *Flatland* in Appendix I.

Reproducibility Statement

The source code of *Flatland*, which generates the protective perturbations, can be found at: https://anonymous.4open.science/r/Flatland-7777. The repository provides a minimal working example, and has been tested using NVIDIA H200/L20/V100 GPUs. Detailed hardware requirements and system setup procedures can be found in the online instructions. We will release research metadata (e.g., .GLB objects) after the publication of this paper.

Ethics Statement

This paper does not include any real human faces or personal photographs in our experiments or demonstrations. The cartoon character shown in the example is generated using GPT-5 based on prompts designed by the authors. Moreover, this paper does not use materials from existing franchises to avoid copyright infringement.

AI Use Statement

The source code of *Flatland* was written by the authors. We used Claude-Opus-4.7 through Claude Code to assist with running experiments, such as managing configuration files, conducting ablation studies, and performing comparisons. The visualization scripts were generated by the LLM, after which we manually tuned the styling parameters. In addition, we used the Claude Code agent to organize the released open-source repository, generating the readme documents and refactoring the source-code structure. The main-text content of this paper was written manually by the authors; the LLM was only used afterwards to correct grammar errors and typos. Meanwhile, we use the LLM to cross-check the described design details against the code implementation to ensure consistency. All diagrams in the paper were drawn manually with vector drawing tools. The dancer illustration in Figure 1, which serves as the example image protected by *Flatland*, is synthesized by GPT-5.

References

Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. SF3D: stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. In *CVPR*, pp. 16240–16250. Computer Vision Foundation / IEEE, 2025.

Bochuan Cao et al. IMPRESS: evaluating the resilience of imperceptible perturbations against unauthorized data usage in diffusion-based generative AI. In *NeurIPS*, 2023.

Angel X. Chang et al. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], 2015.

Rui Chen et al. Dora: Sampling and benchmarking for 3d shape variational auto-encoders. In *CVPR*, pp. 16251–16261. CVG / IEEE, 2025a.

Zhaoxi Chen et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *CVPR*, pp. 26576–26586. CVF / IEEE, 2025b.

Jasmine Collins et al. ABO: dataset and benchmarks for real-world 3d object understanding. In *CVPR*, pp. 21094–21104. IEEE, 2022.

Matt Deitke et al. Objaverse: A universe of annotated 3d objects. In *CVPR*, pp. 13142–13153. IEEE, 2023a.

Matt Deitke et al. Objaverse-xl: A universe of 10m+ 3d objects. In *NeurIPS*, 2023b.

Laura Downs et al. Google scanned objects: A high-quality dataset of 3d scanned household items. In *ICRA*, pp. 2553–2560. IEEE, 2022.

Jacob Feldman. What is a visual object? *Trends in Cognitive Sciences*, 7(6):252–256, 2003.

Hanna Foerster et al. Lightshed: Defeating perturbation-based image copyright protections. In *Security*, pp. 7271–7290. USENIX, 2025.

Fortune Business. 3d gaming technology market size, industry share — forecast [2025-2032], 2026a. URL https://www.fortunebusinessinsights.com/3d-gaming-technology-market-104776.

Fortune Business. 3d printing market size, share & industry analysis report [2025-2032], 2026b. URL https://www.fortunebusinessinsights.com/3d-printing-market-101902.

Huan Fu et al. 3d-future: 3d furniture shape with texture. *Int. J. Comput. Vis.*, 129(12):3313–3337, 2021.

Global Market Insights. 3d digital asset market to cross $72.9 bn by 2032, 2026. URL https://www.globenewswire.com/news-release/2024/12/03/2990378/0/en/3D-Digital-Asset-Market-to-cross-72-9-Bn-by-2032.html.

Google. Google ai studio — gemini 2.5 flash image, 2026. URL https://aistudio.google.com/models/gemini-2-5-flash-image.

Robert Hönig et al. Adversarial perturbations cannot reliably protect artists from generative AI. In *ICLR*, 2025.

John Horvath. Disney pulls star wars models from thingiverse: An inside look at copyright issues in the 3d space. All3DP, 2026. URL https://all3dp.com/disney-pulls-star-wars-models-from-thingiverse/.

Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. arXiv preprint arXiv:2501.12202, 2025a.

Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material. arXiv preprint arXiv:2506.15442, 2025b.

Hunyuan3D Team. Hunyuan3d 2.5: Towards high-fidelity 3d assets generation with ultimate details. arXiv preprint arXiv:2506.16504, 2025c.

Hunyuan3D Team. Hunyuan3d-omni: A unified framework for controllable generation of 3d assets. arXiv preprint arXiv:2509.21245, 2025d.

Kurt Koffka. *Principles of Gestalt psychology*. routledge, 2013.

Thanh Van Le et al. Anti-dreambooth: Protecting users from personalized text-to-image synthesis. In *ICCV*, pp. 2116–2127. IEEE, 2023.

Huiying Li et al. Blacklight: Scalable defense for neural networks against query-based black-box attacks. In *USENIX Security Symposium*, pp. 2117–2134. USENIX Association, 2022.

Weiyu Li et al. Craftsman3d: High-fidelity mesh generation with 3d native diffusion and interactive geometry refiner. In *CVPR*, pp. 5307–5317. CVF / IEEE, 2025a.

Wenhao Li et al. Step1x-3d: Towards high-fidelity and controllable generation of textured 3d assets. arXiv preprint arXiv:2505.07747, 2025b.

Yuxin Li et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. arXiv preprint arXiv:2502.06608, 2025c.

Chumeng Liang and Xiaoyu Wu. Mist: Towards improved adversarial examples for diffusion models. arXiv preprint arXiv:2305.12683, 2023.

Chumeng Liang et al. Adversarial example does good: Preventing painting imitation from diffusion models via adversarial examples. In *ICML*, volume 202, pp. 20763–20786. PMLR, 2023.

Yixin Liu et al. Metacloak: Preventing unauthorized subject-driven text-to-image diffusion-based synthesis via meta-learning. In *CVPR*, pp. 24219–24228. IEEE, 2024.

MarketsAndMarkets. 3d digital asset market surges to $51.8 billion by 2029 - dominated by autodesk (us), siemens (germany), adobe (us), 2026. URL https://www.marketsandmarketsblog.com/3d-digital-asset-market-worth-51-8-billion-by-2029.html.

Weili Nie et al. Diffusion models for adversarial purification. In *ICML*, volume 162, pp. 16805–16827. PMLR, 2022.

Maxime Oquab et al. Dinov2: Learning robust visual features without supervision. *Trans. Mach. Learn. Res.*, 2024, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, pp. 4172–4182. IEEE, 2023.

Chunlin Qiu et al. VOID: Defeating unauthorized mimicry in latent diffusion models. In *Security*. USENIX, 2026.

James M. Rehg. Toys4k dataset, 2026. URL https://rehg.org/publication/dataset2/.

Xuanchi Ren et al. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *CVPR*, pp. 4209–4219. IEEE, 2024.

Robin Rombach et al. High-resolution image synthesis with latent diffusion models. In *CVPR*, pp. 10674–10685. IEEE, 2022.

Hadi Salman et al. Raising the cost of malicious ai-powered image editing. In *ICML*, volume 202, pp. 29894–29918. PMLR, 2023.

Shawn Shan et al. Glaze: Protecting artists from style mimicry by text-to-image models. In *USENIX Security Symposium*, pp. 2187–2204. USENIX, 2023.

Shawn Shan et al. Nightshade: Prompt-specific poisoning attacks on text-to-image generative models. In *SP*, pp. 807–825. IEEE, 2024.

Jiaming Song et al. Denoising diffusion implicit models. In *ICLR*. OpenReview.net, 2021.

Qi Song et al. Geometry cloak: Preventing TGS-based 3d reconstruction from copyrighted images. In *NeurIPS*, 2024.

Michael Tschannen et al. SigLIP 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. arXiv preprint arXiv:2502.14786, 2025.

Peter A Van der Helm. Simplicity versus likelihood in visual perception: from surprisals to precisals. *Psychological Bulletin*, 126(5):770, 2000.

Johan Wagemans et al. A century of gestalt psychology in visual perception: I. perceptual grouping and figure–ground organization. *Psychological bulletin*, 138(6):1172, 2012a.

Johan Wagemans et al. A century of gestalt psychology in visual perception: Ii. conceptual and theoretical foundations. *Psychological bulletin*, 138(6):1218, 2012b.

Aurel Wannig et al. Automatic spread of attentional response modulation along gestalt criteria in primary visual cortex. *Nature Neuroscience*, 14(10):1243–1244, 2011.

Bang Wong. Points of view: Gestalt principles (part 1). *Nature Methods*, 7(11):863, 2010a.

Bang Wong. Points of view: Gestalt principles (part 2). *Nature Methods*, 7(12):941, 2010b.

Shijie Wu et al. Direct3d-s2: Gigascale 3d generation made easy with spatial sparse attention. In *NeurIPS*, 2025.

Tong Wu et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In *CVPR*, pp. 803–814. IEEE, 2023.

Tong Wu et al. Habitat synthetic scenes dataset (hssd), 2026. URL https://3dlg-hcvc.github.io/hssd/.

Zhirong Wu et al. 3d shapenets: A deep representation for volumetric shapes. In *CVPR*, pp. 1912–1920. IEEE Computer Society, 2015.

Jianfeng Xiang et al. Structured 3d latents for scalable and versatile 3d generation. In *CVPR*, pp. 21469–21480. CVF / IEEE, 2025.

Jiale Xu et al. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. arXiv preprint arXiv:2404.07191, 2024.

Haotian Xue et al. Toward effective protection against diffusion-based mimicry through score distillation. In *ICLR*. OpenReview.net, 2024.

Xianghui Yang et al. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation. arXiv preprint arXiv:2411.02293, 2024.

Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4):92:1–92:16, 2023.

Longwen Zhang et al. CLAY: A controllable large-scale generative model for creating high-quality 3d assets. *ACM Trans. Graph.*, 43(4):120:1–120:20, 2024.

Zhengyue Zhao et al. Can protective perturbation safeguard personal data from being exploited by stable diffusion? In *CVPR*, pp. 24398–24407. IEEE, 2024.

Zibo Zhao et al. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *NeurIPS*, 2023.

Jiachen Zhou et al. Dormant: Defending against pose-driven human image animation. In *Security*, pp. 5209–5228. USENIX, 2025.

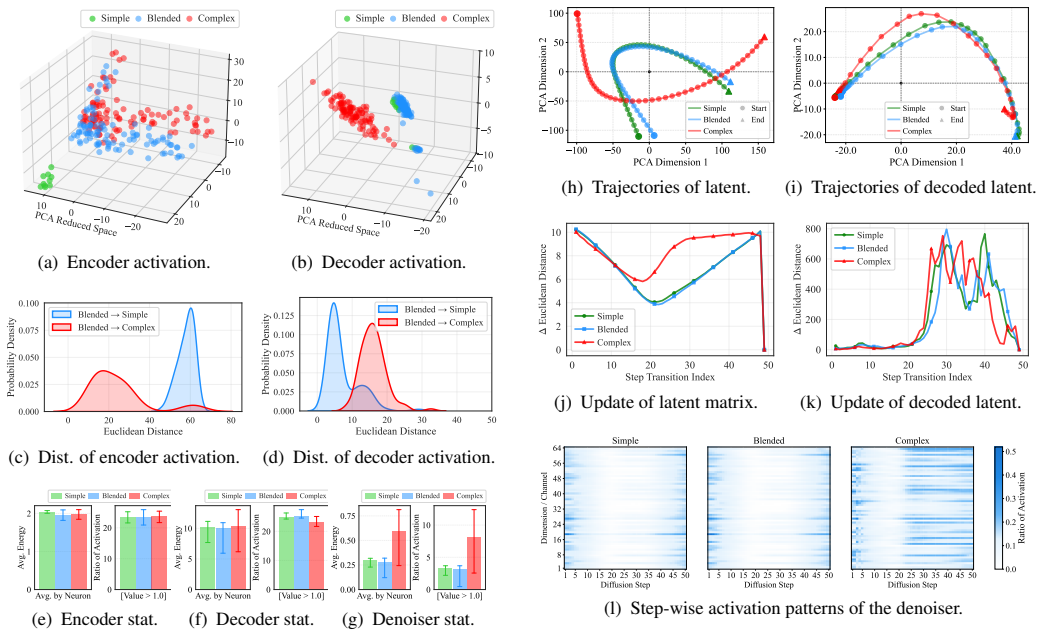
Qingnan Zhou and Alec Jacobson. Thingi10k dataset, 2026. URL https://github.com/Thingi10K/Thingi10K.

Peifei Zhu et al. Watermark-embedded adversarial examples for copyright protection against diffusion models. In *CVPR*, pp. 24420–24430. IEEE, 2024.

A Neurological Mechanism of Simplicity Bias

We perform a neuron activation pattern analysis across the core components of 3D diffusion models, i.e., the vision encoder, denoiser, and 3D decoder. Our findings reveal the reason behind the simplicity bias in 3D diffusion: simple structures inhibit the denoising process. For this analysis, we randomly selected 100 images depicting complex objects (Hunyuan3D Team, 2025b) and individually blended them with 10 simple objects, similar to those used in Figure 3 ($\alpha = 0.95$). Finally, we input the images to the Hunyuan3D-2.1 model and derive the activation amplitude of the last layer of the three components for analysis. In addition, we record the activation patterns of the denoiser and 3D decoder at each step of the generation. We derive following key observations.

First, the vision encoder is not the root cause of the simplicity bias. Figure 10(a) depicts the scatters of neuron activation patterns of vision encoders projected by the PCA algorithm. The activation patterns triggered by images blended with complex and simple objects are adjacent to those of complex objects; the associated Euclidean distance is 2.415 times lower than the distance to the simple objects (see Figure 10(c)). In contrast, Figure 10(b) shows that the corresponding activation pattern of the 3D decoder is close to simple objects with 2.236 times lower distances (see distribution in Figure 10(d)). Comparing the activation patterns of the vision encoder and 3D decoder, the signal of simple objects in the blended images introduces minimal variation in the vision encoder's activation, and thus the vision encoder is not directly related to the preference for reconstructing simple structures.



(a) Encoder activation. (b) Decoder activation. (c) Dist. of encoder activation. (d) Dist. of decoder activation. (e) Encoder stat. (f) Decoder stat. (g) Denoiser stat. (h) Trajectories of latent. (i) Trajectories of decoded latent. (j) Update of latent matrix. (k) Update of decoded latent. (l) Step-wise activation patterns of the denoiser.

Figure 10: Neuron Activation Pattern Analysis for Diffusion Models' Simplicity Bias. (a)-(d) Activation patterns of the encoder and decoder: blending simple structures barely changes the encoder's activation, but noticeably shifts the decoder's patterns toward those of simple objects. (e)-(g) Activation energy and ratio of active neurons: only the denoiser exhibits a significant variance under simple-object blending (Min-Max error bars). (h)-(i) The latent update trajectory shows a *fast-slow-fast* pattern, while the decoded latent shows a *slow-fast-slow* pattern. (j)-(k) Step-wise update widths: simple-structure blending reduces the update width during the mid-to-late diffusion stage. (l) Step-wise activation heatmap of the denoiser: activation is inhibited during the critical mid-to-late stage under simple-structure blending.

Second, variances in neural activation statistics are only observed within the denoiser. Figures 10(e)–10(g) compare two statistics—the average energy of activation and the ratio of neurons in hyperactive status (i.e., activation value higher than 1.0). We observe that the vision encoder and 3D decoder show no significant variance when processing images depicting simple, complex, or blended objects. The differences in average energy associated with the vision encoder and 3D decoder can be bounded by 3.63% and 2.69%; the differences in hyper-activated ratio can be bounded by 1.28% and 7.62% for the two modules, respectively. In contrast, we observe significant variance in the statistics for the denoiser. When processing the complex images, the average energy associated with the 25th step (middle of generation) is 50.45% and 52.80% higher than that of simple and blended objects; the ratio of hyperactive neurons is also 58.64% and 61.52% higher, respectively. The results suggest that the denoiser is the primary component associated with the phenomenon. Next, we dive into the denoiser to analyze its step-wise activations.

Third, inhibition of the mid-to-late denoising stage leads to the neglect of complex structures. We find a slow latent space update and inactive neuron activation during mid-to-late diffusion stages—the stages critical for producing complex structures. Specifically, Figure 10(h) depicts the latent space update trajectory projected by the PCA algorithm, and Figure 10(j) illustrates the step width of the update measured by Euclidean distance. We observe that the trajectory of the latent exhibits a fast-slow-fast update pattern, where the speed of update decreases from the initial steps to around the 20th step, and afterward accelerates in the remaining steps. Such a fast-slow-fast update is entirely different from the slow-fast-slow update associated with the latent decoded by the 3D decoder (Figure 10(i)), which indicates that the mid-to-late stage is critical for generating the 3D geometry. However, during such critical stages ($20 \sim 40$ steps), blending signal of simple objects into the images introduces a $15.18 \sim 49.20$ lower update speed compared with exclusively processing complex objects. Actually, according to Figure 10(l), the neuron network is inactive during the stage due to the presence of the simple object signals; the average ratio of activated neurons is $4.99\% \sim 11.05\%$ lower. Therefore, the simple structures introduce inhibition to neuron activation and latent update in the mid-to-late diffusion steps—the steps critical for reconstruction.

In conclusion, the reason why 3D diffusion models exhibit a bias toward generating simple objects is that simple objects in the visual signal inhibit neuron activation during the mid-to-late denoising stage, ultimately causing the model to fail at producing complex 3D structures and to generate simple ones instead. Therefore, the mode exhibit simplicity bias as suggested by the Law of Prägnanz.

B Design Details of Protection System

This section begins by formalizing *Flatland*'s perturbation task as an optimization problem, then describes the three key components of *Flatland*, which serve as the objects, constraints, and loss functions of the optimization problem.

B.1 Overall Framework Formulation

We first model the three key components of 3D diffusion models: (1) a vision encoder $\mathcal{E}_{\text{img}}(\cdot)$ that extracts image features; (2) a denoiser $\mathcal{F}_{\text{diff}}(\cdot)$ that progressively denoises latent 3D representations conditioned on the encoded image features; and (3) a structural decoder $\mathcal{D}_{\text{3D}}(\cdot)$ that reconstructs the final 3D structure. The complete generation model can be formulated as:

$$\mathcal{M}(\mathbf{x}) = \mathcal{D}_{\text{3D}}(\mathcal{F}_{\text{diff}}(\mathbf{Z}_{\text{3D}}|\mathcal{E}_{\text{img}}(\mathbf{x}))), \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$ is the input image, $\mathcal{E}_{\text{img}}(\mathbf{x})$ provides the conditioning signal, $\mathbf{Z}_{\text{3D}} \in \mathbb{R}^{D_1\times D_2\times\cdots\times D_n}$ represents the high-dimensional 3D latent tensor (the total dimensionality $\prod_{i=1}^{n} D_i \gg H\times W\times 3$), and $\mathcal{M}(\mathbf{x})$ is the reconstructed 3D structure.

Idealized Objective. Given an original image $\mathbf{x}_{\text{src}}$, the ideal objective is to generate a protected image $\mathbf{x}_{\text{prot}} = \mathbf{x}_{\text{src}} + \boldsymbol{\delta}$ that maximizes the distance between 3D reconstructions:

$$\max_{\boldsymbol{\delta}} \; d_{\text{3D}}(\mathcal{M}(\mathbf{x}_{\text{src}}), \mathcal{M}(\mathbf{x}_{\text{prot}})), \tag{2}$$

where $d_{\text{3D}}(\cdot,\cdot)$ is a 3D distance metric (e.g., Chamfer Distance) and $\boldsymbol{\delta} \in \mathbb{R}^{H\times W\times 3}$ represents the perturbation. However, this objective is intractable in practice because the specific forms and parameters of $\mathcal{E}_{\text{img}}(\cdot)$, $\mathcal{F}_{\text{diff}}(\cdot)$, and $\mathcal{D}_{\text{3D}}(\cdot)$ are unknown under our assumptions.

***Flatland*'s Objective.** To address the intractability under the model-agnostic assumption, we reformulate the optimization task by operating in a 2D feature space rather than targeting the 3D reconstruction pipeline directly. Specifically, for a given original image $\mathbf{x}_{\text{src}}$, we formulate the calculation of $\boldsymbol{\delta}$ as a constrained optimization that balances protection effectiveness and visual quality:

$$\begin{aligned} \min_{\boldsymbol{\delta}} \quad & \mathcal{L}(\mathbf{x}_{\text{src}} + \boldsymbol{\delta}; \{\mathbf{x}_{\text{tgt}}^{(i)}\}_{i=1}^{K}), \\ \text{s.t.} \quad & \|\boldsymbol{\delta}\|_\infty \le \epsilon \cdot \mathbf{W}(\mathbf{x}_{\text{src}}), \end{aligned} \tag{3}$$

where $\mathcal{L}(\cdot; \{\mathbf{x}_{\text{tgt}}^{(i)}\}_{i=1}^{K})$ is the loss function defined over a set of K target images containing simple structures, ϵ controls the maximum perturbation amplitude, and $\mathbf{W}(\mathbf{x}_{\text{src}}) \in [0,1]^{H\times W}$ is the placement constraint of amplitude.

The reformulated optimization consists of three key components that work together to generate perturbations: (1) *VLM-Guided Simple Structure Synthesis* determines the optimization targets $\{\mathbf{x}_{\text{tgt}}^{(i)}\}_{i=1}^{K}$ by creating images with simple structures (Section B.2); (2) *Line-Art-Based Perturbation Placement* defines the constraint through a weight function $\mathbf{W}(\mathbf{x}_{\text{src}})$ that limits the amplitude (Section B.3); and (3) *ViT-Based Perturbation Refinement* serves as the optimization method with loss functions $\mathcal{L}(\cdot)$. It generates perturbations satisfying both the objectives and constraints (Section B.4). With the optimization targets, constraints, and loss functions determined by these three components, *Flatland* can solve the perturbations $\boldsymbol{\delta}$.

B.2 VLM-Guided Simple Structure Synthesis

We leverage a VLM to generate images with simple structures as the optimization objectives. The images satisfy two requirements: (i) they contain simple structures to mislead diffusers into producing the simple structures; (ii) they maintain visual similarity to the original images, ensuring they lie

within a small neighborhood of the originals in feature space. The proximity allows low-amplitude perturbations to inject simple structures by bridging the distance.

VLM-Based Photograph Generation. To satisfy these requirements, *Flatland* instructs the VLM to paint the original image on flat paper, place the paper inside a translucent cylindrical container, and generate a photograph of the container. This approach fulfills both requirements simultaneously. First, the generated photograph visually resembles the original image, as it captures the painted original content. Second, the flat paper and regular cylinder serve as the simple structures to be injected. Formally, we transform the original image $\mathbf{x}_{\text{src}}$ into a target image $\mathbf{x}_{\text{gen}}$ as a photograph with realistic lighting and shadow effects. The generation process is defined as: $\mathbf{x}_{\text{gen}} = \mathcal{G}_{\text{VLM}}(\mathbf{x}_{\text{src}}, \mathcal{P})$, where $\mathcal{G}_{\text{VLM}}$ is the VLM and $\mathcal{P}$ is the text prompt:

"Please create a realistic photograph showing:
1. Draw this image on a paper, fill the paper.
2. The paper is placed inside a cylindrical sealed container with a lid on top.
3. The container should be translucent frosted glass.
4. Zoom in so that the container fills most of the frame.
5. Use realistic lighting and shadows to make it look like a real photograph."

Note that the prompt requires the painted original image to fill the paper and uses hints of *sealed* and *lid* to ensure that the image occupies the central region of the container, thereby improving visual similarity. Moreover, we specify the container to be *translucent frosted glass*, which occludes the paper to suppress true structural information. Finally, we zoom the scene and add *lighting and shadow effects* to accentuate the deceptive structures of these simple objects.

Salient Region Detection. To further improve visual similarity between the original image and the generated photograph, we identify salient regions in both images, in order to align them in the next step. First, we extract the foreground regions from both images using background removal $\mathcal{R}_{\text{bg}}(\cdot)$. For the original image $\mathbf{x}_{\text{src}}$, we obtain the foreground mask $\mathbf{M}_{\text{src}} = \mathcal{R}_{\text{bg}}(\mathbf{x}_{\text{src}})$. To account for potential boundary uncertainties, we apply morphological dilation to expand the foreground mask by d pixels: $\mathbf{M}_{\text{src}}^{\text{dilated}} = \text{Dilate}(\mathbf{M}_{\text{src}}, d)$:

$$\text{Dilate}(\mathbf{M}, d)(y, x) = \begin{cases} 1 & \text{if } \exists (y', x') : \mathbf{M}(y', x') = 1 \\ & \quad \wedge \|(y - y', x - x')\|_2 \leq d \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Similarly, for a generated photograph $\mathbf{x}_{\text{gen}}$, we remove the background to obtain $\mathbf{M}_{\text{gen}} = \mathcal{R}_{\text{bg}}(\mathbf{x}_{\text{gen}})$. Then, we identify the paper region within the container by detecting the foreground bounding box $\mathbf{B}_{\text{fg}}$ and applying inward padding p to exclude the edges. Let $\mathbf{B}_{\text{fg}} = [x_{\min}, y_{\min}, x_{\max}, y_{\max}]$ denote the bounding box, e.g., $x_{\min} = \min\{x \mid \exists\, y \in [0, H], \mathbf{M}_{\text{gen}}(y, x) > 0\}$. The inward padding operation produces the paper region bounding box; we then extract the paper mask by cropping:

$$\mathbf{M}_{\text{paper}}(y, x) = \begin{cases} \mathbf{M}_{\text{gen}}(y, x) & \text{if } x \in [x_{\min} + p, x_{\max} - p] \\ & \quad \wedge y \in [y_{\min} + p, y_{\max} - p] \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Finally, we extract the paper content by applying the mask: $\mathbf{x}_{\text{paper}} = \mathbf{x}_{\text{gen}} \odot \mathbf{M}_{\text{paper}}$, where $\odot$ denotes Hadamard multiplication that broadcasts the binary mask across RGB channels.

Salient Region Registration. Since the VLM-generated photograph may exhibit a different scale and position compared to the original image, we perform affine registration to align the paper content with the original foreground. We parameterize the transformation as a 2×3 affine matrix:

$$\mathbf{\Theta} = \begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \end{bmatrix}, \tag{6}$$

where $(s_x, s_y) \in [1.0, +\infty)$ are scale factors and $(t_x, t_y) \in \mathbb{R}^2$ are translation offsets in pixels.

We define the affine warping operator $\mathcal{A}(\mathbf{I}, \mathbf{\Theta})$ that applies transformation $\mathbf{\Theta}$ to image $\mathbf{I}$ as:

$$\mathcal{A}(\mathbf{I}, \mathbf{\Theta})(y, x) = \mathbf{I}(s_y y + t_y, s_x x + t_x), \tag{7}$$

where pixel values at non-integer coordinates are obtained via bilinear interpolation. The registration jointly optimizes foreground overlap, feature matching, and transformation regularity:

$$\begin{aligned}\boldsymbol{\Theta}^* = \arg\max_{\boldsymbol{\Theta}} \quad & \text{IoU}(\mathbf{M}_{\text{src}}^{\text{dilated}}, \mathcal{A}(\mathbf{M}_{\text{paper}}, \boldsymbol{\Theta})) \\ & + \lambda_f \cdot \text{FeatureMatch}(\mathbf{x}_{\text{src}}, \mathcal{A}(\mathbf{x}_{\text{paper}}, \boldsymbol{\Theta})) \\ & - \lambda_\Theta \cdot \left\| \boldsymbol{\Theta} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \right\|_F^2,\end{aligned} \tag{8}$$

where $\text{IoU}(\cdot,\cdot)$ computes the intersection-over-union between foreground masks, $\text{FeatureMatch}(\cdot,\cdot)$ measures feature point matches using the ORB algorithm, and $\|\cdot\|_F$ denotes the Frobenius norm. The Frobenius norm term $\|\boldsymbol{\Theta} - \mathbf{I}_{2\times 3}\|_F^2$ penalizes deviations from the identity transformation, to obtain a transformation with minimal rescaling and translation. After obtaining the optimal transformation $\boldsymbol{\Theta}^*$, we apply it to the entire foreground to produce the final registered target image: $\mathbf{x}_{\text{tgt}} = \mathcal{A}(\mathbf{x}_{\text{gen}} \odot \mathbf{M}_{\text{gen}}, \boldsymbol{\Theta}^*)$.

Visualization Analysis. Figure 11(a) presents the distribution of feature vectors associated with original, generated, and aligned images, as well as reference images containing only cylinders. We use the same images as those used in Section 3, and the associated feature vectors are extracted using a ResNet-50 model (projected via PCA). We find that the cosine distance between the reference images and the original images is 1.439 times greater than that between our synthesized images and the original images, because the synthesized images contain paper drawings of the original image to improve similarity (see Figure 11(b)). Moreover, alignment further reduces the cosine distance to the original image in the visual feature space by 15.33%. This allows *Flatland* to apply low-amplitude perturbations to inject simple structural information by steering the original images toward images with simple structures.

B.3 Line-Art-Based Perturbation Placement

This module computes a weight matrix $\mathbf{W}(\mathbf{x}_{\text{src}})$ that controls the maximum perturbation amplitude at each pixel. Since structural information is primarily conveyed through edges and contours in images, we place perturbations on these critical regions while reducing perturbation magnitude elsewhere to preserve visual quality.

Line Art Generation and Binarization. We generate a line art based on the source image using the VLM:

$$\mathbf{x}_{\text{line}} = \mathcal{G}_{\text{VLM}}(\mathbf{x}_{\text{src}}, \text{"Generate line art."}). \tag{9}$$

Next, the generated line art $\mathbf{x}_{\text{line}} \in \mathbb{R}^{H\times W\times 3}$ is converted to grayscale. It is then binarized using Otsu's thresholding method to obtain a binary line mask $\mathbf{L} \in \{0,1\}^{H\times W}$:

$$\mathbf{L}(y,x) = \begin{cases} 1 & \text{if } \text{Gray}(\mathbf{x}_{\text{line}}(y,x)) > \tau_{\text{Otsu}} \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\text{Gray}(\cdot)$ converts RGB to grayscale and τ_{Otsu} is the threshold computed by Otsu's method. Note that the pixels near lines are critical regions containing structural information, as line art primarily depicts significant depth changes.

Density and Distance Analysis. We analyze both the spatial distribution and local density of lines to identify critical regions requiring stronger perturbations.

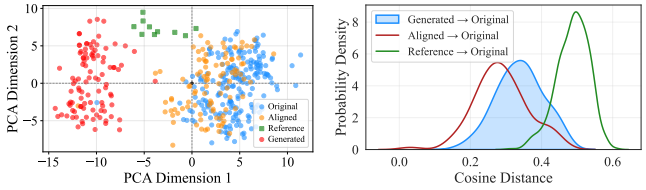
First, to identify regions with dense line structures, we compute the local line density using an averaging kernel. The density map $\mathbf{C} \in [0,1]^{H\times W}$ is defined as:

$$\mathbf{C}(y,x) = \frac{1}{\pi r^2} \sum_{(y',x')\in\mathcal{N}_r(y,x)} \mathbf{L}(y',x'), \tag{11}$$

where $\mathcal{N}_r(y,x) = \{(y',x') : (y'-y)^2 + (x'-x)^2 \le r^2\}$ is a circular neighborhood of radius r centered at (y,x). The density-based weight matrix $\mathbf{W}_{\text{dens}} \in [0,1]^{H\times W}$ is directly given by $\mathbf{W}_{\text{dens}} = \mathbf{C}$.

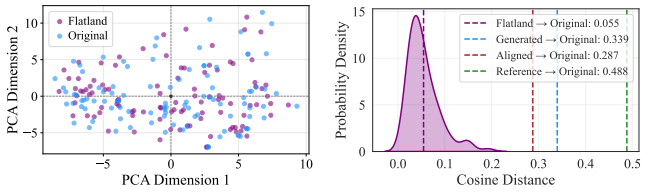
Second, we compute the Euclidean Distance transform to measure each pixel's distance to the nearest line:

$$\mathbf{D}(y,x) = \min_{(y',x')\in\mathcal{L}} \sqrt{(y-y')^2 + (x-x')^2}, \tag{12}$$



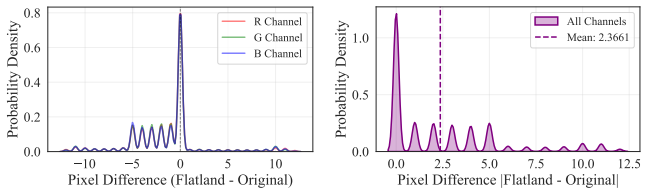
(a) Visual feature distribution. (b) Distance distribution.

Figure 11: Distance of generated and aligned images with simple structures.



(a) Visual feature distribution. (b) Distance distribution.

Figure 12: Distance of *Flatland*-protected and original images.



(a) Amplitude by channels. (b) Amplitude distribution.

Figure 13: Amplitude analysis for *Flatland*'s protective perturbation.

where $\mathcal{L} = \{(y, x) : \mathbf{L}(y, x) = 1\}$ is the coordinate set of all pixels on the line art, and $\mathbf{D} \in \mathbb{R}_{\geq 0}^{H \times W}$ is the distance map. Afterward, we normalize the distance map and apply a smooth decay function to compute distance-based weights. The normalized distance $\mathbf{D}_{\text{norm}} \in [0, 1]^{H \times W}$ is denoted by:

$$\mathbf{D}_{\text{norm}}(y, x) = \min\left(\frac{\mathbf{D}(y, x)}{d}, 1\right). \tag{13}$$

The distance-based weight matrix $\mathbf{W}_{\text{dist}} \in [0, 1]^{H \times W}$ is computed by applying an accelerated cosine decay function: $\mathbf{W}_{\text{dist}}(y, x) = f_{\text{decay}}(\mathbf{D}_{\text{norm}}(y, x))$, where:

$$f_{\text{decay}}(t) = \left[\frac{1 + \cos(\pi t)}{2}\right]^2, \quad t \in [0, 1]. \tag{14}$$

This decay function ensures smooth weight transitions and reduces perceptibility.

Weight Computation. We combine distance and density weights to form the base weight matrix: $\mathbf{W}_{\text{base}} = (\mathbf{W}_{\text{dist}} + \mathbf{W}_{\text{dens}})/2$ ($\mathbf{W}_{\text{base}} \in [0, 1]^{H \times W}$). In addition, we consider the importance indicated by foreground/background regions; we compute an adaptive minimum weight matrix by applying a sigmoid function to the foreground mask:

$$\mathbf{W}_{\text{min}} = w_{\text{bg}} + (w_{\text{fg}} - w_{\text{bg}}) \cdot \sigma_k(\mathbf{M}_{\text{src}}), \tag{15}$$

where $\sigma_k(m)$ is a sigmoid function, and w_{fg} and w_{bg} are the weights for the foreground and background ($w_{\text{fg}} + w_{\text{bg}} = 1$).

The final weight matrix $\mathbf{W} \in [0, 1]^{H \times W}$ combines the base weight with the adaptive minimum:

$$\mathbf{W} = \mathbf{L} + (1 - \mathbf{L}) \odot [\mathbf{W}_{\text{min}} + (1 - \mathbf{W}_{\text{min}}) \odot \mathbf{W}_{\text{base}}]. \tag{16}$$

It ensures that pixels near lines receive high perturbation scaled by distance and density, with smooth transitions between pixels that improve imperceptibility.

B.4 VIT-BASED FEATURE SPACE NOISE OPTIMIZATION

This module optimizes the protective perturbation to minimize the distance to target images while satisfying the constraints. Our optimization approach is characterized by three key features: (1) *Condensed ViT feature space*; (2) *Multi-target optimization*; and (3) *Scale dynamism*.

Condensed ViT Feature Space. The strong representation ability of the ViT model enables searching for perturbations that can effectively represent simple structures. However, the scale of attention-like feature matrices extracted by the ViT is significantly larger than that of traditional feature vectors, which introduces optimization difficulty and computational overhead. Thus, we compress the feature space using sequential pooling to obtain condensed representations.

For an input image $\mathbf{x} \in \mathbb{R}^{H\times W\times 3}$, the pretrained ViT encoder $\mathcal{E}_{\text{ViT}}(\cdot)$ outputs a feature matrix:

$$\mathbf{F} = \mathcal{E}_{\text{ViT}}(\mathbf{x}) = [\mathbf{f}_{\text{cls}}; \mathbf{f}_{\text{patch}}] \in \mathbb{R}^{(P+1)\times d_{\text{ViT}}}, \tag{17}$$

where $\mathbf{f}_{\text{cls}} \in \mathbb{R}^{1\times d_{\text{ViT}}}$ is the global-summary token that encodes semantic information of the entire image, taken as the CLS token when the backbone provides one and as the attention-pooling output otherwise. $\mathbf{f}_{\text{patch}} \in \mathbb{R}^{P\times d_{\text{ViT}}}$ is the patch token matrix where each row corresponds to a patch, P is the number of patches, and d_{ViT} is the feature dimension.

To condense the features, we apply sequential average pooling to the patch tokens along the token dimension. We divide the P patch tokens into $\lfloor \frac{P}{k} \rfloor$ groups of k consecutive tokens, and compute the average within each group:

$$\mathbf{F}^{(i)}_{\text{pooled}} = \frac{1}{k} \sum_{j=(i-1)k+1}^{ik} \mathbf{f}^{(j)}_{\text{patch}}, \quad i = 1, 2, \ldots, \lfloor \frac{P}{k} \rfloor. \tag{18}$$

Let $\mathbf{F}_{\text{pooled}} \in \mathbb{R}^{\lceil \frac{P}{k} \rceil \times d_{\text{ViT}}}$ denote the matrix of all pooled patch tokens. We then concatenate the CLS token with the pooled patch tokens and flatten the matrix:

$$\bar{\phi}_{\text{ViT}}(\mathbf{x}) = \text{Flatten}([\mathbf{f}_{\text{cls}}; \mathbf{F}_{\text{pooled}}]) \in \mathbb{R}^{(1+\lceil \frac{P}{k} \rceil)\cdot d_{\text{ViT}}}. \tag{19}$$

We then apply L2 normalization to obtain the final feature representation: $\phi_{\text{ViT}}(\mathbf{x}) = \bar{\phi}_{\text{ViT}}(\mathbf{x})/\|\bar{\phi}_{\text{ViT}}(\mathbf{x})\|_2$.

In the feature space, we measure the distance using a metric combining cosine and Euclidean distances to leverage their complementary strengths. Given two images encoded by the ViT feature extractor $\phi_{\text{ViT}}(\cdot)$, we define the distance between their features $\mathbf{f}_1 = \phi_{\text{ViT}}(\mathbf{x}_1)$ and $\mathbf{f}_2 = \phi_{\text{ViT}}(\mathbf{x}_2)$ as:

$$d_{\text{ViT}}(\mathbf{f}_1, \mathbf{f}_2) = \left(1 - \frac{\mathbf{f}_1 \cdot \mathbf{f}_2}{\|\mathbf{f}_1\|_2 \|\mathbf{f}_2\|_2}\right) + \frac{\|\mathbf{f}_1 - \mathbf{f}_2\|_2}{\|\mathbf{f}_1\|_2 + \|\mathbf{f}_2\|_2}. \tag{20}$$

Multi-Target Optimization. To create diverse optimization objectives, we generate N candidate photographs using different random seeds but the same prompts as described in Section B.2. We then select the top-K candidates that are most similar to the original image based on the distance:

$$\{\mathbf{x}^{(i)}_{\text{tgt}}\}_{i=1}^{K} = \text{Top}\left(\{\mathbf{x}^{(j)}_{\text{gen}}\}_{j=1}^{N}; K, -d_{\text{ViT}}(\phi_{\text{ViT}}(\mathbf{x}_{\text{src}}), \phi_{\text{ViT}}(\cdot))\right). \tag{21}$$

This selection process further improves the randomness of the perturbation and reduces discernible patterns. The target distance loss is the minimum distance to the selected targets:

$$\mathcal{L}_{\text{target}} = \min_{i\in\{1,\ldots,K\}} d_{\text{ViT}}(\phi_{\text{ViT}}(\mathbf{x}_{\text{prot}}), \phi_{\text{ViT}}(\mathbf{x}^{(i)}_{\text{tgt}})). \tag{22}$$

Scale Dynamism Enhancement. To improve robustness against image transformations, we apply random scale augmentation at every iteration during optimization. At each iteration, we randomly scale the perturbed image by a factor $s \sim \mathcal{U}(1-\Delta_s, 1+\Delta_s)$, then resize it back to the original dimensions before feature extraction:

$$\tilde{\mathbf{x}}^{(t)}_{\text{prot}} = \text{Resize}(\text{Resize}(\mathbf{x}^{(t)}_{\text{prot}}, s^{(t)} \cdot (H, W)), (H, W)), \tag{23}$$

where $\text{Resize}(\cdot, \text{size})$ denotes bilinear interpolation. In addition, we combine the target distance loss with ℓ_2 regularization $\mathcal{L}_{\text{reg}} = \|\boldsymbol{\delta}\|_2^2$ to further reduce the amplitude. The complete loss function at iteration t is:

$$\begin{aligned} \mathcal{L}^{(t)}_{\text{total}} &= \lambda_{\text{reg}} \mathcal{L}^{(t)}_{\text{reg}} + \mathcal{L}^{(t)}_{\text{target}} \\ &= \lambda_{\text{reg}} \|\boldsymbol{\delta}^{(t)}\|_2^2 + \min_{i\in\{1,\ldots,K\}} d_{\text{ViT}}(\phi_{\text{ViT}}(\tilde{\mathbf{x}}^{(t)}_{\text{prot}}), \phi_{\text{ViT}}(\mathbf{x}^{(i)}_{\text{tgt}})). \end{aligned} \tag{24}$$

Table 2: Comparison of typical protection methods against diffusion models.

Method	Configuration			Prior Knowledge				Surrogate Requirement			Robustness			
	Model	Targeted Task	Stage	w/o Param	w/o Dataset	w/o Arch.	Black Box	w/o Surr. Dataset	w/o Surr. Model	w/o Surr. Query	Scale	Crop	Filter	Purify
Nightshade	T-to-I	Concept Drawing	Train	✓	✗	✓	✗	✓	✗	✓	-	-	-	✗
Zhu et al.	TI-to-I	Instructed Drawing	Test	✗	✓	✗	✗	✗	✓	✗	-	-	✓	-
MetaCloak	T-to-I	Personalization	Finetune	✓	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗
Mist	I-to-I	Style Transfer	Train	✗	✓	✗	✗	✓	✓	✗	-	-	-	✓
Glaze	I-to-I	Style Transfer	Train	✗	✓	✗	✗	✓	✗	✓	-	-	-	✗
AdvDM	TI-to-I	Style Transfer	Test	✗	✓	✗	✗	✓	✓	✗	-	-	✓	-
Anti-D.B.	T-to-I	Personalization	Finetune	✗	✗	✗	✗	✗	✗	✗	-	-	✓	-
PhotoGuard	TI-to-I	Image Editing	Test	✗	✗	✗	✗	✓	✓	✗	-	✗	✗	-
SDS	TI-to-I	Image Editing	Test	✗	✗	✗	✗	✓	✓	✗	-	-	-	-
Dormant	I-to-V	Human Motion Gen.	Test	✓	✓	✓	✓	✗	✗	✗	-	-	✓	✓
VOID	I-to-I	Unauth. Mimicry	Test	✗	✓	✗	✗	✓	✗	✗	-	-	✓	✓
Flatland	I-to-3D	3D Structure Gen.	Test	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

[1] T-to-I: Text-to-Image, I-to-I: Image-to-Image, TI-to-I: Text-Image-to-Image, I-to-V: Image-to-Video. And - indicates not investigated.

Overall Procedure. The optimization iteratively updates the perturbation $\boldsymbol{\delta}^{(t)} = \mathbf{x}_{\text{prot}}^{(t)} - \mathbf{x}_{\text{src}}$ via the following three steps:

1. *Gradient Computation.* Compute the gradient of the total loss with respect to the perturbed image: $\mathbf{g}^{(t)} = \nabla_{\mathbf{x}_{\text{prot}}} \mathcal{L}_{\text{total}}^{(t)}$.
2. *Gradient Update.* Update the perturbed image using the Adam optimizer with learning rate α: $\hat{\mathbf{x}}_{\text{prot}}^{(t+1)} = \text{Adam}(\mathbf{x}_{\text{prot}}^{(t)}, \mathbf{g}^{(t)}, \alpha)$.
3. *Applying Constraints.* Clip the perturbation and add it to the image $\mathbf{x}_{\text{prot}}^{(t+1)} = \mathbf{x}_{\text{src}} + \boldsymbol{\delta}^{(t+1)}$, where

$$\boldsymbol{\delta}^{(t+1)} = \text{Clip}\Big(\hat{\mathbf{x}}_{\text{prot}}^{(t+1)} - \mathbf{x}_{\text{src}},\ -\epsilon \odot \mathbf{W}(\mathbf{x}_{\text{src}}),\ \epsilon \odot \mathbf{W}(\mathbf{x}_{\text{src}})\Big). \tag{25}$$

Perturbation Analysis. From Figure 12, we can observe that *Flatland* adds minimal perturbation to preserve visual similarity, where the associated average cosine distance between the features is 0.055 (i.e., close to zero, high similarity). Moreover, Figure 13 presents channel- and pixel-wise analysis of the perturbation amplitude. From Figure 13(a), we find that *Flatland* evenly distributes the perturbation across RGB channels without significant variance to avoid discernible patterns. Meanwhile, the perturbation's average amplitude is only 2.366, which ensures the protection preserves visual usability, as *Flatland* only places the perturbation on 55.68% of pixels (Figure 13(b)).

C Comparison with Related Protection Methods

Table 2 compares *Flatland* with existing protection methods against diffusion models across four dimensions: configuration (target model, task, and defense stage), required prior knowledge, resource cost, and robustness. Unlike prior work that targets 2D visual semantics (concept drawing, style transfer, personalization, image editing, human motion) under white-box or grey-box settings, *Flatland* targets *3D structural* reconstruction, requires no model parameters, datasets, architectures, or surrogates, and remains robust to rescaling, cropping, filtering, and purification. Please refer to Section 6 for the discussion of these methods.

D Details of Experimental Setup

D.1 Implementation

We prototype *Flatland* with ∼2,700 lines of Python 3.10 code. Noise optimization is implemented in PyTorch 2.4 with CUDA 12.4. We use Gemini 2.5 Flash Image (a.k.a. NanoBanana (Google, 2026)) as the VLM and SigLIP 2 (Tschannen et al., 2025) (So400m/16-256, ∼428M vision-tower parameters) as the ViT image encoder. All qualitative examples are exported in GLB format and rendered in Blender 4.5 using the CYCLES engine. Experiments run on NVIDIA H200 GPUs. Appendix D.5 illustrates the settings of hyperparameters.

D.2 Targeted 3D Reconstruction Models

We select ten AI models that are either industry-endorsed or accepted by major academic venues within the past two years, with parameters ranging from 0.8 to 5.3 billion. These models adopt different latent-space representations, including voxel-associated latent vectors (Wu et al., 2025; Xiang et al., 2025; Chen et al., 2025b), global latent vectors representing the entire 3D space (Hunyuan3D Team, 2025b; Li et al., 2025b;a), and triplane-based latent vectors (Boss et al., 2025; Li et al., 2025c). The majority of the selected methods use Transformer backbones to implement diffusion (i.e., DiT (Peebles & Xie, 2023)), trained with flow matching or rectified flow objectives. In addition, to demonstrate the generalizability of *Flatland* beyond diffusion, we further include non-diffusion Transformer-based 3D reconstruction models, e.g., SF3D (Boss et al., 2025). Note that we exclude traditional non-Transformer GAN-based methods (Zhang et al., 2023) and outdated approaches that require multi-view inputs (Yang et al., 2024; Xu et al., 2024). Additionally, we disable computationally intensive colored texture generation for the public models, as this study focuses on the reconstructed structures.

D.3 Structure Datasets

We use 11 public datasets that can be categorized into four groups (cleaning procedure can be found in Appendix D.6):

- **Scanned Objects:** Real-world objects captured using 3D scanning technologies, providing high-fidelity structures of household items (Downs et al., 2022) and natural objects (Wu et al., 2023).
- **CAD Modeling:** Structures designed by Computer-Aided Design tools, e.g., indoor scenes (Wu et al., 2026), furniture (Fu et al., 2021), industry products (Collins et al., 2022), and 3D-printable structures (Zhou & Jacobson, 2026).
- **Community Artworks:** Large-scale collections of diverse 3D objects created by online communities, characterized by varied artistic styles (Rehg, 2026) and qualities (Deitke et al., 2023a;b).
- **Traditional Benchmarks:** Datasets primarily used for 3D shape classification and segmentation tasks, with objects represented as point clouds (Chang et al., 2015) or meshes (Wu et al., 2015).

In addition, we evaluate on the ten datasets curated by the ten public models, each containing various complex objects.

D.4 Evaluation Metrics

We measure the effectiveness using three categories of widely-used metrics:

- **Point Cloud Metrics:** Point clouds are sampled and evaluated using Chamfer Distance (CD), F-Score with precision/recall at threshold $\tau = 0.01$, and PSNR for overall geometric accuracy.
- **Feature Space Metrics:** These metrics evaluate perceptual quality by rendering meshes from $N_v = 24$ viewpoints and computing LPIPS similarity (AlexNet-based perceptual loss), CLIP similarity (CLIP ViT-B/32), and DINOv2 similarity (Giant version).
- **Spatial Metrics:** These metrics assess volumetric consistency via Surface IoU (S-IoU) with threshold $\tau_s = 0.01$ and Volumetric IoU (V-IoU) at resolution $r = 128^3$.

To quantify *Flatland*'s effectiveness, we compute the change in each metric relative to the ground-truth structures: $\Delta_{\text{metric}} = \text{Metric}(\mathcal{M}_{\text{prot}}, \mathcal{M}_{\text{GT}}) - \text{Metric}(\mathcal{M}_{\text{src}}, \mathcal{M}_{\text{GT}})$. All meshes undergo preprocessing including non-visible face removal, quadric decimation to 50K faces, and ICP-based alignment. Point clouds are uniformly sampled with 10K points for geometric metrics.

D.5 Hyperparameters

We tune the hyperparameters using images from Hunyuan3D-2.1 as the validation set, with the objective of maximizing the increase in CD. All loss-related weights (λ_f, λ_Θ, λ_{reg}) are adjusted to achieve similar magnitudes across different loss terms, ensuring balanced contributions during optimization. Moreover, pixel-related hyperparameters (e.g., d and r) are manually tuned by visually inspecting whether the output images satisfy the goal of the corresponding module. Optimizer-related ones are determined through grid search by fixing $N = 1$ and $K = 1$. Finally, we gradually increase N and K until no further improvement is observed.

Table 3: Hyperparameters and default settings.

Notation	Default	Description
Module 1: VLM-Guided Simple Structure Synthesis		
N	5	Number of candidate photographs generated.
K	3	Number of top similar candidates selected.
d	25 pixels	Dilation radius and inward padding.
λ_f	1×10^{-3}	Weight for ORB feature matching in registration.
λ_Θ	1.0	Regularization weight.
Module 2: Line-Art-Based Perturbation Placement		
r	30 pixels	Radius for local line density computation.
w_{bg}	0.4	Weight for background regions.
Module 3: ViT-Based Perturbation Optimization		
ϵ	0.2	Overall amplitude factor.
α	1×10^{-2}	Learning rate for Adam optimizer.
T_{max}	1000	Maximum number of optimization iterations.
λ_{reg}	1×10^{-4}	Regularization weight for ℓ_2 norm penalty.
Δ_s	0.2	Scale range for random scaling augmentation.
k	10	Sequential pooling size for ViT patch tokens.

D.6 Datasets Cleaning Pipeline

As the quality of large-scale artwork datasets is low, we use a preprocessing pipeline similar to that used in existing studies (Xiang et al., 2025). To generate the images as inputs for AI models, we render the images of original objects (10^2 per dataset) using Blender's CYCLES engine, with ambient and lighting parameters identical to those of Trellis (Xiang et al., 2025). The 3D object is rendered from four distinct camera viewpoints at $[0, \frac{\pi}{2}, \pi, \frac{3\pi}{4}]$ to obtain four potential views. To establish the canonical front view, we first leverage Rembg to remove the background of each rendered image. Following existing practices (Deitke et al., 2023b), the front view is then selected by identifying the image that maximizes the combined metric of foreground object coverage and the mean normalized entropy of the RGB color channels' histograms. Afterward, the selected image is centered and padded with a 15% border (Hunyuan3D Team, 2025b). Finally, the image is scaled to a standard resolution of 518×518 pixels, which is the default input size for existing models.

E Additional Quantitative Results

Additional Datasets and Metrics. Figure 14 reports the relative reconstruction quality degradation on curated datasets: *Flatland* achieves 224.27% average CD degradation and 51.17% F-Score degradation, with the strongest effect concentrated on each model's own examples (diagonal elements in the heatmap). Figure 15 further reports the absolute reconstruction quality on curated datasets, showing that *Flatland* remains effective even when the AI models reach their best performance. Table 5 provides per-method F-Score degradation across all public datasets, and Table 6 extends the cross-metric analysis to point-cloud, feature-space, and spatial metrics.

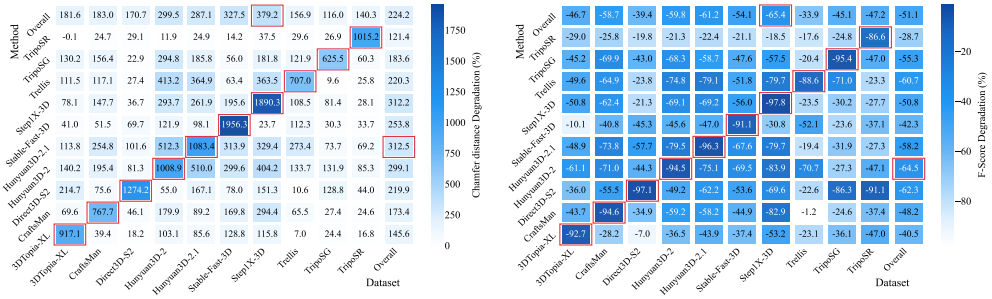
Details of Baseline Comparison. Table 4 reports the per-model Chamfer Distance degradation of ten representative 2D image protection methods on the ten target 3D models: existing methods achieve only -0.15% to 8.13% average CD degradation, whereas *Flatland* reaches 195.01%, outperforming the best baseline by 178.36% on average. For every protected image, the target 3D reconstruction model is invoked with a freshly sampled random seed. All baselines follow their original default configurations, with the noise budget set to $8/255$, which is larger than the average amplitude used by *Flatland*'s default configuration (see Appendix B). Note that all baselines compared here operate under the black-box setting and target diffusion architectures. Moreover, we can only compare model-independent methods, and thus exclude methods designed for one particular model architecture, e.g., the watermark for a triplane space (Song et al., 2024).

Visual Quality. We further quantify the visual change introduced by *Flatland*'s perturbation between the clean input and the protected input. Across the 11 public datasets, the protected images retain high fidelity: PSNR ranges from 37.36 to 38.64 dB (average 37.89 dB) and SSIM ranges from 0.9276 to 0.9447 (average 0.9352), indicating that the perturbation is imperceptible in standard image-quality terms.

Table 4: Chamfer Distance Degradation: Compare with other copyright protection against diffusion models.

Methods	3DTopia	CraftsMan	Direct3D-S2	HY3D-2	HY3D-2.1	SF3D	Step1X-3D	Trellis	TripoSG	TripoSR	Avg.	Deg.
NightShade	-0.50	**0.70**	-4.50	4.40	5.90	19.40	-0.40	-3.50	2.30	**0.50**	2.43	-192.58↓
Zhu *et al.*	0.60	11.80	**-0.20**	2.60	8.90	13.20	3.30	-1.00	-3.90	14.80	5.01	-190.00↓
MetaCloack	-1.20	-5.30	5.40	-7.60	**2.80**	18.00	0.30	1.30	**0.00**	8.20	2.19	-192.82↓
Mist	-1.80	23.00	1.00	1.30	6.60	18.70	-0.30	-2.20	1.90	-16.00	3.22	-191.79↓
Glaze	1.40	0.90	-7.00	**0.90**	42.20	8.50	-1.70	-1.30	3.30	-15.70	3.15	-191.86↓
AdvDM	0.40	22.10	5.40	-6.50	53.00	18.20	-0.40	-2.00	6.10	-15.00	8.13	-186.88↓
Anti-D.B.	**0.20**	1.50	-4.00	-7.40	8.60	-9.10	4.50	0.90	0.40	5.00	**0.06**	-194.95↓
PhotoGuard	2.00	45.90	-0.70	-7.70	8.00	2.00	**0.10**	**-0.20**	0.80	3.30	5.35	-189.66↓
SDS	-0.50	50.40	-9.50	-5.20	6.00	-12.90	3.80	-6.30	7.10	8.00	4.09	-190.92↓
Dormant	0.60	-29.80	8.00	2.20	5.20	**-0.00**	**-0.10**	2.90	5.20	9.70	0.39	-194.62↓
Flatland	69.14	231.92	64.12	351.61	282.45	142.51	243.02	197.24	257.09	111.00	195.01	-191.61↓
Min Imp.	67.14↑	181.52↑	56.12↑	347.21↑	229.45↑	123.11↑	238.52↑	194.34↑	249.99↑	96.20↑	178.36↑	N/A

[1] **Bold** indicates the best result in each column, underline indicates the worst.
[2] Blue↑ indicates improvement (higher is better), Red↓ indicates degradation (lower is worse).



(a) Chamfer distance degradation.

(b) F-Score degradation.

Figure 14: Relative 3D reconstruction quality degradation indicates the effectiveness of Flatland protection on curated datasets.

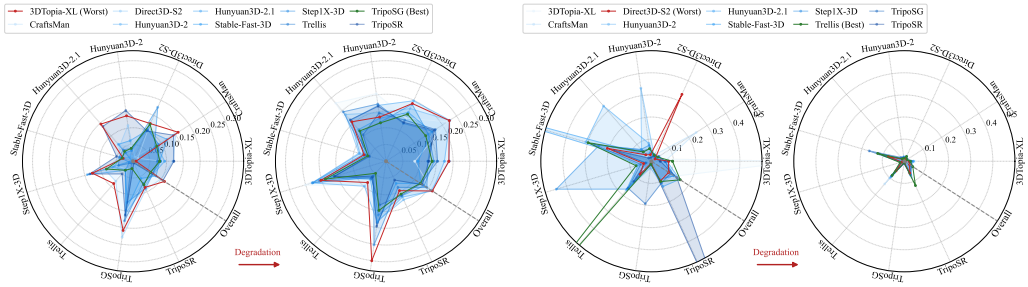
F Additional Qualitative Results

We provide additional visual comparisons across diverse datasets and AI models, covering scanned objects, CAD components, community artworks, and traditional benchmarks. Figure 17 extends the public-dataset examples shown in the main text, and Figure 16 shows the results on the curated datasets. In addition, Figures 18 and 19 visualize the worst-case triplanes of Hunyuan3D-2.1 and Trellis under *Flatland*'s protection. Across all examples, *Flatland* consistently prevents the AI models from reconstructing the original sensitive structures. We observe that most low-degradation cases stem from poor baseline reconstruction quality rather than protection failure. When original generation already suffers from substantial geometric errors (high baseline CD), the absolute CD increase introduced by *Flatland* becomes less pronounced.

G Details of Ablation Study

Key Designs by Modules. Figure 8 shows that simply synthesizing paper or a container yields a 2.19∼5.01% CD increase, and random noise (Gaussian/Uniform) only 0.06∼3.22%; even blank-paper synthesis (132.62%) and random placement (145.91%) underperform our full design, confirming that synthesis and placement are jointly essential. The per-module ablations further isolate each component: disabling registration collapses protection effectiveness to a 5.35% CD increase, and alternative VLMs underperform ours; for optimization, removing cosine distance (8.13%), disabling compression (4.09%), or using a single synthesized object (131.06%) similarly hurts effectiveness; for placement, edge-only fails catastrophically (3.95%), and distance-only (145.65%) or dense-only (150.51%) are both suboptimal, with the combined metric achieving the best result.

Different Visual Spaces. We further replace the ViT encoder in the optimization module with several alternative encoders and observe comparable protection effectiveness. When *Flatland* uses DINOv3-Large, the average CD degradation reaches 205.14% with a 44.79% F-Score decrease; with AIMv2-Huge, *Flatland* achieves 185.21% CD degradation and 37.56% F-Score decrease; and with



(a) Chamfer distance degradation.

(b) F-Score degradation.

Figure 15: Absolute 3D reconstruction quality degradation indicates the effectiveness of Flatland protection on curated datasets.

Table 5: F-Score relative degradation (%) on public datasets (lower is better).

Methods	Scanned Objects		CAD Modeling				Community Artworks			Benchmark		Avg.
	GSO	Omini.3D	HSSD	3D-FUT.	ABO	Thingi10K	Obj.-XL	Objaverse	Toys4k	ModelNet	ShapeNet	
3DTopia-XL	-52.13	-32.53	-48.17	-31.00	-28.36	-38.03	-46.54	-61.49	-73.44	-40.16	-61.27	-46.65
CraftsMan	-67.21	-37.89	-23.42	-74.98	-46.27	-58.08	-46.86	-43.22	-37.83	-13.95	-90.54	-49.11
Direct3D-S2	-50.87	-36.73	-48.33	-22.02	-12.90	-44.15	-48.90	-36.37	-23.94	-43.62	21.20	-31.51
Hunyuan3D-2	-60.95	-56.17	-76.80	-41.80	-55.73	-56.35	-50.78	-47.58	-53.23	-60.89	-60.80	-56.46
Hunyuan3D-2.1	-55.90	-58.77	-70.41	-46.02	-31.49	-9.14	-60.59	-35.18	-77.65	-44.06	-35.99	-47.75
Stable-Fast-3D	-42.12	-24.94	-8.40	-40.46	-23.40	-49.60	-17.48	-54.15	-71.68	-41.05	-26.74	-36.37
Step1X-3D	-50.79	-54.92	-71.67	-32.41	-56.85	-26.62	-49.17	-40.21	-66.92	-24.20	-43.32	-47.01
Trellis	-35.90	-49.56	-16.85	-52.07	-32.01	-31.62	-65.22	-63.30	-60.42	-40.78	-23.46	-42.84
TripoSG	-60.78	-56.76	-60.75	-61.69	-46.51	-49.86	-1.44	-53.39	-69.38	-15.62	-44.43	-47.33
TripoSR	-87.56	-29.47	-33.32	-38.74	-36.66	-38.82	-43.19	-15.78	-50.58	-55.18	-42.02	-42.85
Average	-56.42	-43.77	-45.81	-44.12	-37.02	-40.23	-43.02	-45.07	-58.51	-37.95	-40.74	-44.79

*The table highlights the best accuracy in ● and the worst accuracy in ●.

MetaCLIP 2 ViT-H/14, *Flatland* achieves 235.05% CD degradation and 40.51% F-Score decrease. All three results are on par with the 195.01% CD / 44.79% F-Score reported by the default ViT. Note that none of these encoders coincides with the vision encoder used by any of the ten target 3D reconstruction models, so the numbers directly validate the black-box assumption (i.e., no access to the internal encoder of the target model). When *Flatland* happens to share the same encoder family as the target model—for instance, using DINOv2-Giant to match Hunyuan3D-2.1's vision encoder—the protection reaches 198.56% CD degradation and 46.69% F-Score decrease, only marginally above the black-box results. Overall, *Flatland* can effectively produce the protective perturbation when applying various visual encoders, while it prevents 3D diffusion models with different encoder architectures from unauthorized reconstruction.

Decoy Structures. We also analyze the choice of injected structures. First, steering the optimization toward an unrelated random target image—another object sampled from the same dataset—is unable to effectively interfere with reconstruction: CD degradation is bounded within 10.25% and F-Score degradation within 5.32%. Second, replacing the cylindrical decoy with other simple 3D structures produces comparable protection: injecting cubes, cones, and hexahedra as decoys yields 178.51%, 169.24%, and 182.91% CD degradation, respectively. This is consistent with Figure 3, which verifies that a broad family of simple-structure signals can trigger the model's simplicity bias.

Activation Pattern Analysis. Finally, we conduct a neuron-activation analysis on the model when it processes *Flatland*-protected images. Feeding the protected images into Hunyuan3D-2.1, the resulting activation patterns closely resemble those observed when the model reconstructs simple objects (Figure 10). Specifically, compared to processing unprotected images, the average activation energy of the denoiser is 48.76% lower around the 25th diffusion step, and the average ratio of activated neurons in steps 20–40 is 5.65%–12.72% lower. These findings mirror the activation patterns triggered by simple-structure inputs analyzed in Appendix A, confirming that *Flatland*'s perturbation triggers the same inhibitory dynamics inside the denoiser.

Table 6: Performance of *Flatland*'s protection measure in different metrics.

Method	Point Cloud Metrics						Feature Space Metrics						Spatial Metrics			
	CD ↑		F-Score ↓		PSNR ↓		LPIPS ↑		CLIP ↓		DINO ↓		S-IoU ↓		V-IoU ↓	
	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.	Abs.	Rel.
3DTopia-XL	+0.027	+69.14	-0.087	-35.18	-7.189	-23.87	+0.074	+34.36	-0.031	-3.74	-0.097	-29.48	-0.044	-26.14	-0.010	-50.21
CraftsMan	+0.042	+231.92	-0.121	-49.11	-8.342	-25.98	+0.087	+53.57	-0.064	-7.34	-0.197	-35.21	-0.090	-42.53	-0.037	-38.80
Direct3D-S2	+0.027	+64.12	-0.052	-31.51	-6.233	-24.96	+0.065	+27.92	-0.038	-4.56	-0.091	-22.76	-0.030	-27.25	-0.003	-42.67
HY3D-2	+0.058	+351.61	-0.160	-56.46	-11.660	-36.21	+0.128	+93.15	-0.089	-10.13	-0.299	-50.84	-0.125	-58.72	-0.035	-36.01
HY3D-2.1	+0.050	+282.45	-0.148	-47.75	-10.499	-33.27	+0.104	+70.16	-0.094	-10.78	-0.304	-53.73	-0.112	-46.68	-0.041	-40.28
SF-3D	+0.031	+142.51	-0.092	-36.37	-5.273	-16.16	+0.052	+33.44	-0.053	-6.11	-0.210	-45.84	-0.067	-37.09	-0.026	-11.71
Step1X-3D	+0.050	+243.02	-0.126	-47.01	-10.072	-32.41	+0.099	+60.62	-0.079	-9.11	-0.262	-47.86	-0.097	-51.17	-0.037	-39.17
Trellis	+0.044	+197.24	-0.098	-42.84	-9.234	-30.14	+0.078	+46.05	-0.072	-8.32	-0.308	-52.12	-0.070	-45.72	-0.035	-38.58
TripoSG	+0.048	+257.09	-0.143	-47.33	-10.276	-32.29	+0.119	+77.44	-0.067	-7.67	-0.238	-42.76	-0.108	-51.71	-0.041	-41.49
TripoSR	+0.026	+111.00	-0.051	-42.85	-2.962	-7.60	+0.046	+30.32	-0.033	-3.91	-0.141	-36.54	-0.044	-48.82	-0.020	-25.67
Average	+0.040	+195.01	-0.108	-43.64	-8.174	-26.29	+0.085	+52.70	-0.062	-7.17	-0.215	-41.71	-0.079	-43.58	-0.029	-36.46

[1] For each metric, Abs. shows absolute change, Rel. shows relative percentage change.
[2] Blue indicates best method, red indicates worst method per column. ↑ means higher is better, ↓ means lower is better.

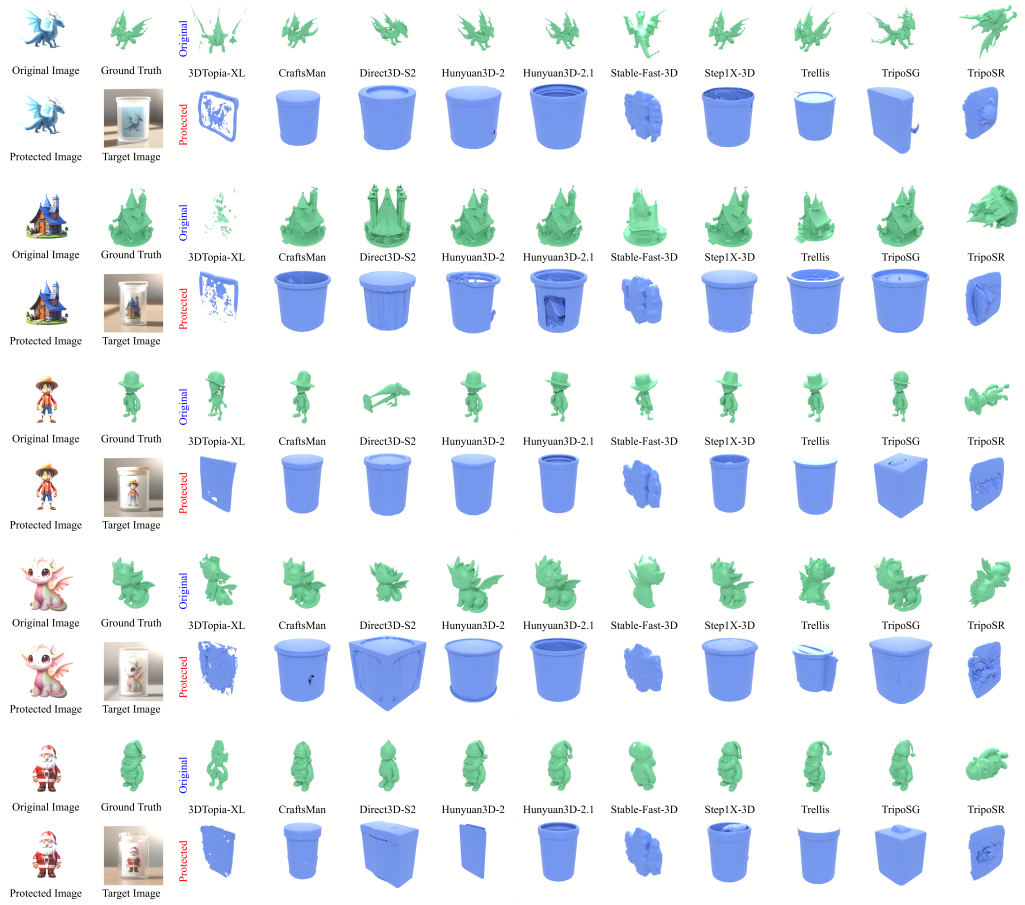
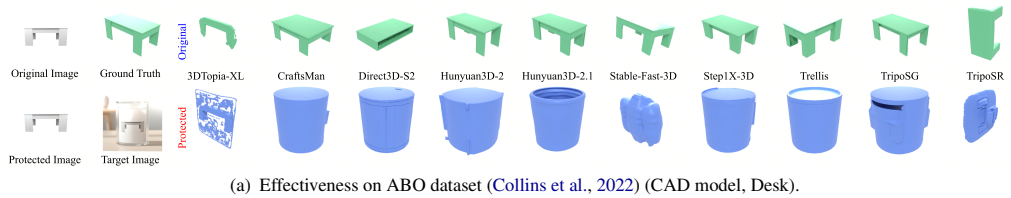


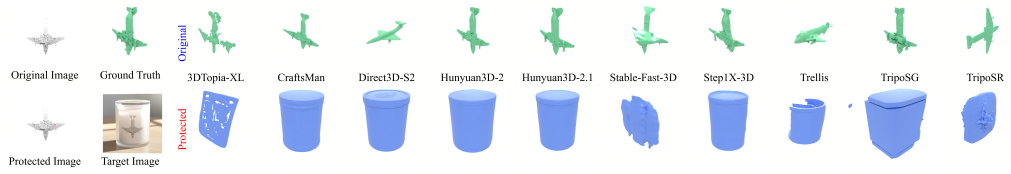
Figure 16: Reconstructed structures with and without the protection of *Flatland* on curated datasets.

H Efficiency Evaluation

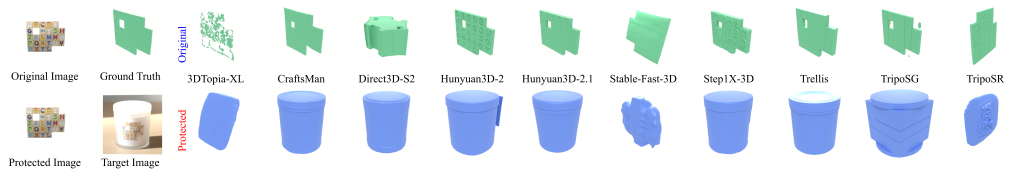
Performance across Datasets. Figure 20(a) presents *Flatland*'s processing time across four dataset categories. *Flatland* demonstrates consistent performance across different data types, achieving 171.96s on CAD models, 172.83s on community artworks, 174.06s on scanned objects, and 173.38s on traditional benchmarks. The minimal variance in processing time indicates that *Flatland*'s computational cost is largely independent of object complexity, texture characteristics, or geometric



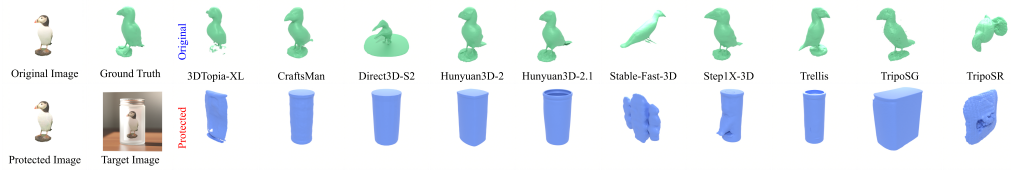
(a) Effectiveness on ABO dataset (Collins et al., 2022) (CAD model, Desk).



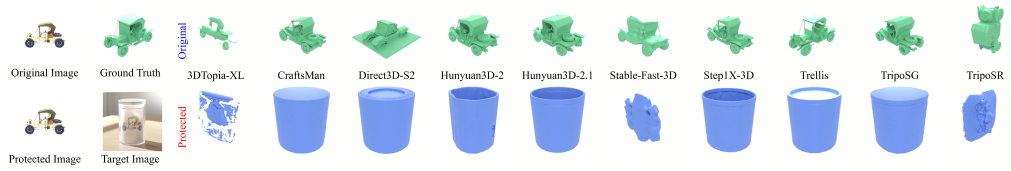
(b) Effectiveness on ShapeNet dataset (Wu et al., 2015) (traditional benchmark, air plane).



(c) Effectiveness on GSO dataset (Downs et al., 2022) (scanned object, toy).



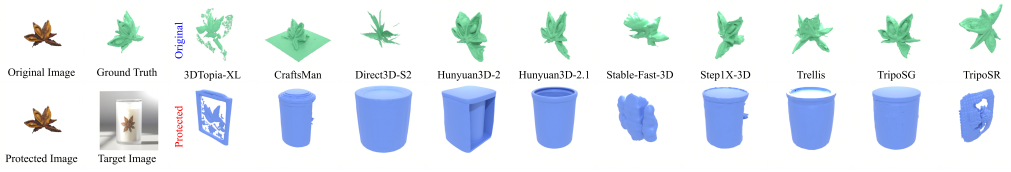
(d) Effectiveness on Objeverse-XL dataset (Deitke et al., 2023b) (artwork, bird).



(e) Effectiveness on Objeverse dataset (Deitke et al., 2023a) (artwork, car).



(f) Effectiveness on Toys4K dataset (Rehg, 2026) (scanned object, bicycle).

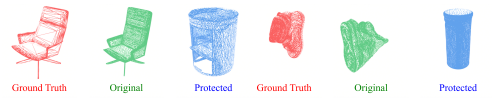


(g) Effectiveness on OminiObject3D dataset (Wu et al., 2023) (scanned object, star anise spice).

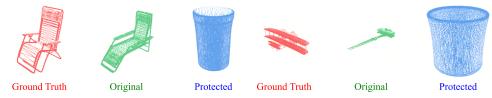
Figure 17: Additional comparison of reconstruction w/ and w/o the protection of *Flatland* on public datasets.

properties. This consistency stems from the fixed optimization iterations in the optimization module, which dominate overall latency.

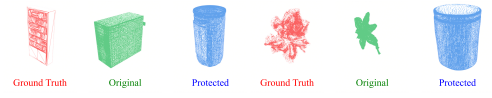
Impact of Image Resolution. Figure 20(b) analyzes how image resolution affects processing time. As expected, high resolutions require more processing time than low resolutions: from 173.65s at 512×512 to 90.82s at 256×256, representing a 47.7% reduction. The optimization module accounts for the majority of this decrease, dropping from 137.80s to 54.07s (−60.8%), as feature extraction and optimization complexity scale with pixel count. Notably, performance at 320×320 shows a



(a) 3D-FUTURE (office chair). (b) GSO (boot).



(c) ABO (folding lounge chair). (d) Toys4K (glider).

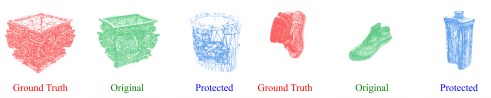


(e) ModelNet (bookshelf). (f) OminiObject3D (spice).

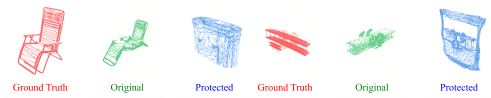


(g) Objaverse-XL (bronze vessel). (h) Thingi10K (ornament).

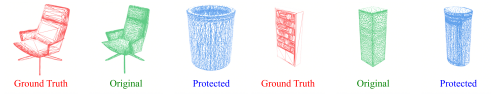
Figure 18: Triplane examples (Hunyuan3D-2.1).



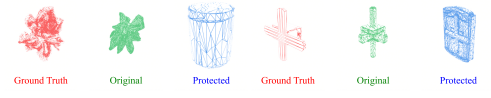
(a) Obj.-XL (bronze vessel). (b) GSO dataset (Boot).



(c) ABO (folding lounge chair). (d) Toys4K dataset (glider).

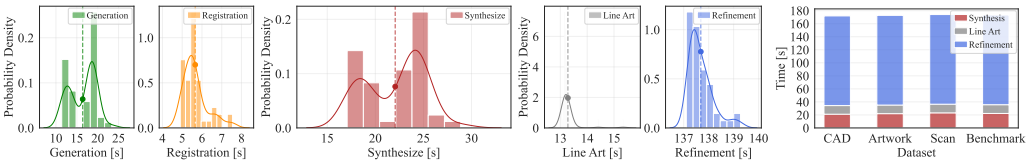


(e) 3D-FUTURE (office chair). (f) ModelNet (bookshelf).



(g) OminiObject3D (spice). (h) Thingi10K (ornament).

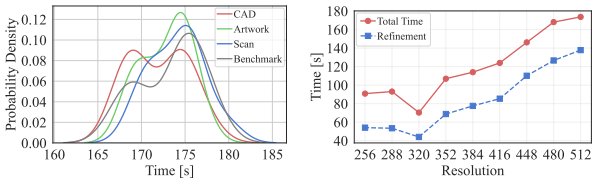
Figure 19: Triplane examples (Trellis).



(a) Generation and registration. (b) Synthesize module overall. (c) Line art and optimization. (d) Latency comparison.

Figure 21: Composition of *Flatland*'s processing latency.

lower processing time, which may be caused by a caching mechanism in the CUDA kernel. For practical deployment, users can improve the speed by processing at lower resolutions (e.g., 384×384 achieves 113.95s with a 34.4% speedup).



(a) Different datasets. (b) Different resolution.

Figure 20: Impacts of dataset and resolution on performance.

Latency Composition by Modules. Figure 21 breaks down *Flatland*'s processing time across three main modules. Figure 21(a) shows that the synthesis module consists of VLM generation (16.41s) and registration alignment (5.66s), totaling 22.07s on average. Figure 21(b) further investigates the synthesis module's overall consistency with minimum 17.07s and maximum 28.87s across all datasets. Figure 21(c) reveals that the optimization module dominates overall latency at 137.69s on average, accounting for approximately 79.5% of total processing time, while line art placement requires only 13.28s. The optimization module's high computational cost stems from iterative feature extraction and gradient-based optimization across the entire image. Figure 21(d) illustrates the latency composition across four dataset groups: synthesis contributes 21.09∼23.13s (12.1∼13.3%), line art placement contributes 13.20∼13.40s (7.6∼7.7%), and optimization contributes 137.63∼137.73s (79.2∼79.9%).

I Limitations and Future Work

Adversarial Training for 3D Diffusion. We conducted a preliminary study on adversarial training. Specifically, we applied *Flatland*'s protective perturbations to 100 example images for Hunyuan3D-2.1 and fine-tuned the model on these images for 10,000 steps under the default training configuration. The fine-tuned model still fails to reconstruct 3D structures, with the CD increase remaining at

191.13%. We attribute this resilience to the fact that *Flatland* exploits the simplicity bias inherent in 3D diffusers—a fundamental property that may not be easily overridden by fine-tuning.

Transferability. *Flatland*'s protection is model-agnostic, i.e., the protective perturbation generated for one image transfers across many models, as shown in Section 5.1. In addition, we observe transferability across images: directly applying the perturbation generated for one image to another still achieves effective protection, with a 154.54% average CD increase on CAD datasets. Such transferability could be further exploited to improve the efficiency of *Flatland*.

Training DNNs with Perturbation. As reported in Section 5.3, LightShed (Foerster et al., 2025), which learns *Flatland*'s perturbation patterns via an autoencoder, fails to accurately remove the perturbations. This is because *Flatland*'s multi-target optimization, random scaling, and VLM-based synthesis jointly inject randomness that prevents any consistent pattern from being learned. Nevertheless, we acknowledge that *Flatland* must continue to evolve against future removal methods beyond those evaluated in our experiments.

Quantitative Analysis for Proprietary Models. Due to generation quotas and the substantial time cost of each request, we primarily conduct qualitative analysis of *Flatland*'s effectiveness on proprietary AI models (Section 5.2, Figure 7(a)). As a small-scale quantitative complement, we submit 10 images sampled from the Hunyuan3D-2.1 curated dataset in both their original and *Flatland*-protected forms to commercial models whose access limits are relatively relaxed. *Flatland* induces a 241.4% CD increase on Meshy-4 and a 201.1% CD increase on Hitem3D v1.0, consistent with the open-source results in Figure 14.

Camera Recapture. We used an iPhone 12 Pro Max to recapture the *Flatland*-protected images from Figure 1, both from a display and from printed paper. The recaptured images still fail to be reconstructed into accurate 3D structures, with CD increases of 186.01% and 154.29%, respectively. This implies that the built-in denoising in cameras struggles to remove the perturbations. We are happy to evaluate any additional removal methods suggested by the audiences.