ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888843

Dual-Process Watermarked Diffusion: Integrating Watermarking With Denoising in Point Clouds

Jinfu Wei[1*], Heng Chang[3*], Xiaohang Liu[2*],Li Liu[3], Likun Li[3], Shiji Zhou[1],
Chengyuan Li[3], Di Xu[3], Wei Gao[2] , Ran Liao[1]
[1]*Tsinghua University, China*
[2]*Peking University, China*
[3]*Huawei Technologies Co., Ltd., China*

***Abstract*—The integration of depth sensing and laser scanning technologies has propelled point cloud data to the forefront of 3D graphical modeling. This paper addresses a critical gap in the literature: the protection of intellectual property in generating point clouds using Diffusion Models (DMs). We introduce Dual-Process Watermarked Diffusion (DPWD), a pioneering watermarking framework for Diffusion Models (DMs) used in point cloud generation. DPWD embeds watermarks directly into DMs, ensuring strong protection against intellectual property theft. To do so, we introduce a two-stage watermark strategy: 1) watermark embedding using a permutation invariant module, and 2) watermark integration into the DM's noise predictor. The framework is robust against common attacks and preserves the quality of generated point clouds. Empirical results on various point cloud tasks demonstrate DPWD's effectiveness in safeguarding intellectual property rights without compromising model performance. DPWD sets a new standard for model protection in the GenAI era.**

***Index Terms*—point cloud, generative models, watermark.**

I. Introduction

As the technological landscape evolves, point cloud data has become a pivotal element in 3D graphical modeling. Significant strides have been made in developing analytical methods for point cloud data, such as classification and segmentation [1]–[5]. At the same time, the learning models for the point cloud generation, particularly those based on Diffusion Models (DMs) [6]–[8], have shown remarkable strength in unsupervised representation learning and garnered increasing research interest in recent years. Motivated by the denoising process in DMs, there have been widespread applications in point cloud generation [9]–[15]. However, compliance issues related to their use, such as model provenance and copyright infringement, remain overlooked. Once a user obtains a point cloud model trained by the model owner, it becomes challenging to trace the origins of the generated results. This poses a significant hurdle for the broad deployment of DM-based point cloud generation models.

Existing invisible watermarking methods of point clouds are centered on post-processing ways. They usually watermark one object in blind [16]–[18] or non-blind [19]–[23] way by employ regular embedding techniques, which makes them vulnerable to attacks if the embedding strategy is known. Furthermore, the lack of differentiability hinders the usage of powerful deep-learning models for watermark embedding and retrieval, a key advantage for robust and secure watermarking. In contrast, deep hiding for images has made great advancements, several works could hide bits, QR codes, and general pictures in a single image [24]–[27]. To protect the image copyright from generative models, a naive way is to provide an API only and append a post-processing watermark to the generated images. Some approaches also take watermarks as input and generate watermarked data directly [28], [29]. Those kinds of watermark approaches can also be introduced into point cloud generative models. However, in the case of model plagiarism, where the pre-trained generative model is obtained by an unauthorized user, these methods will fail by simply removing the post-processing module. As a result, the generative models' watermarking is needed, a way to root watermarks in generative models with only their weights changed. With deep hiding techniques in images, models' watermarking are also well-developed with the help of deep hiding techniques [30]–[35]. Most of them fine-tune a one-step generative model or LDM decoder, which is unsuitable for point cloud DMs. In summary, the main challenges are: 1) traditional watermarking methods, mostly post-processing, cannot be directly integrated into the end-to-end training of DMs; 2) limited by gradient accumulation, existing models' watermark methods designed for single-step forward models cannot be applied on multi-step denoising DMs directly.

To overcome these challenges, we introduce ***D***ual-***P***rocess ***W***atermarked ***D***iffusion on Point Cloud generation (DPWD), the first watermarking framework for point cloud generation tasks in DMs. DPWD is a two-stage model watermarking method that breaks free from the constraints of post-processing watermarking. Inspired by deep 3D mesh watermark algorithm [36], [37] and deep hiding framework HiDDeN [25], we propose a permutation-invariant watermark module (PI-HiDDeN), consisting of a point cloud watermark encoder and corresponding extractor to enhance the robustness of the 3D structure against common watermark attacks such as rotation and downsampling. In the second stage, we introduce a dual-period distilling method to embed a certain watermark into DMs, which bypasses the impact of gradient accumulation due to the iterative denoising process, ensuring the quality of DMs-generated point clouds post-watermarking.

Our experiments across multiple point cloud generation tasks demonstrate the effectiveness of DPWD in watermarking point cloud generation, the maintenance of model functionality post-watermarking, and the robustness of our watermarks against various attacks. In summary, our contributions are as follows:

- To the best of our knowledge, DPWD is the first approach to add watermarks in DMs for point cloud generation, which is important for protecting developers' intellectual property rights.
- DPWD is composed of two stages: PI-HiDDeN tackles the challenge of extracting watermarks in the point cloud from the permutation invariant embedding space, and the dual-period distilling confronts the challenge of the accumulation of gradients in DMs that prevents the completion of the point cloud.
- Extensive experiments on real-world point cloud generation tasks demonstrate that DPWD could effectively implant water-

*Equal Contribution; Corresponding authors.

This research was funded by National Natural Science Foundation of China (62275141 and 62401327); China Postdoctoral Science Foundation (GZB20230358, 2023M741966 and 2024T170464); Hainan Development Project of Science and Technology (ZDYF2022SHFZ323); Shenzhen's S&T Project for Sustainable Development (KCXST20221021111405013).

marks into DMs without downgrading the quality of generated point clouds while preserving robustness to various attacks.

II. Problem Statement

The burgeoning field of diffusion models for point cloud generation faces a significant challenge: **model plagiarism**. Considering the following scenario:

- An individual, Alice, trains a powerful diffusion model to generate intricate 3D point clouds.
- Bob acquires Alice's pre-trained model, either through legitimate or illegitimate means.
- Bob utilizes Alice's model to generate novel point clouds.
- Bob then claims ownership of these generated point clouds, falsely attributing them to his own efforts.

This scenario exemplifies the critical issue of copyright protection for creators who invest significant resources in training these complex models. Traditional approaches to digital rights management are inadequate for point cloud data.

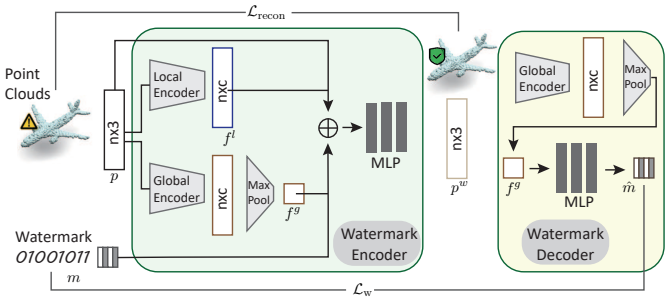

Fig. 1. Architecture of PI-HiDDeN. All the local and global encoders mentioned in this figure share the same permutation-invariant architecture(1-layer DGCNN) to extract features of point clouds.

III. Proposed Method: DPWD

DPWD targets root watermarks in diffusion generation models with only its weights changed so that iterative denoising diffusion models will synthesize watermarked point clouds with invisible changes. We begin with a well-trained watermark module for point clouds and implicitly embed the watermark encoder into the noise predictor in a distilling way. There are two challenging problems for our watermarking pipeline: 1) a differential and robust watermark model for point clouds and 2) an adapted distillation process to effectively embed a watermark in iterative diffusion models. In practice, we utilize DGCNN to construct our watermark models, which naturally respect the permutation invariance of point clouds. In the distillation stage, we propose a novel dual-period watermark distilling method that can effectively embed a watermark in diffusion models for a substantial improvement in safety with negligible impact on the synthesized results.

A. Watermark Module: PI-HiDDeN

1) Architecture: The architecture is shown in Figure 1. We train our watermark encoder $\mathcal{W}_E$ with a matching watermark decoder $\mathcal{W}_D$, the former aims to hide a watermark in the cloud points invisibly, and the latter tries to extract the hidden watermark. In practice, $\mathcal{W}_E$ takes a point cloud p and a k-bits watermark $m \in \{0,1\}^k$, and produces a watermarked point cloud p^w by applying tiny displacements $\epsilon(p,m)$ on points:

$$p^w = \mathcal{W}_E(p,m)_i = \epsilon(p,m)_i + p_i. \tag{1}$$

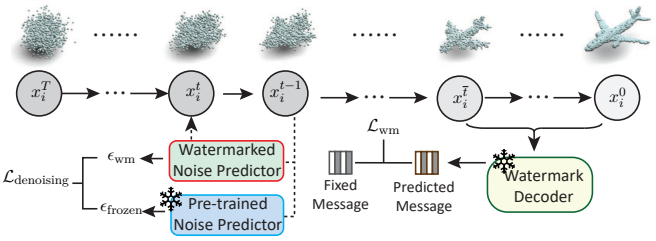

Fig. 2. Framework of dual-period distilling. In the denoising period, $\mathcal{L}_{\text{denoising}}$ ensures the ability to denoising and generate consistency. In the watermarking period, an extra $\mathcal{L}_{\text{wm}}$ is introduced by our pre-trained watermark decoder to integrate watermarks in diffusion models.

The key to embedding watermarks in cloud points is to obtain proper displacements from its local and global information in $\mathcal{W}_D$ and $\mathcal{W}_E$. Intuitively, the local information allows networks to introduce a watermark in its adjacent structure, and the global information makes the watermark more robust when some points are attacked. Specifically, we apply DGCNN as our backbone, which firstly clusters all the points into several groups as the source of our local features f^l and the global features f^g are from the same architecture with only an extra max pooling layer. Then, an MLP layer maps the point p_i with its f_i^l, shared f^g, and watermark m as its corresponding displacement. With respect to $\mathcal{W}_D$, only a global feature is used in our settings.

2) Attack layer: We introduce some classical attack methods in our training stage to improve the robustness. In our attack layer, one of them is randomly applied on the outputs of $\mathcal{W}_E$, which are fed into $\mathcal{W}_D$ with original point clouds. Besides, in the following distilling process, noisy point clouds will be used for extracting watermarks, so it's essentially that the extractor is robust to Gaussian noise. In our attack layer, there are three main attack methods: random zero-means displacements Gaussian noise, downsampling by randomly setting to 0, and rotation on x, y, and z axes. We provide the definition of them as follows:

a) Gaussian noise: This attack introduces random zero-means displacements to the points within a point cloud. The variance σ of Gaussian distribution reflects the degree of attack. It involves adding random values drawn from a Gaussian distribution to each point's coordinates (x, y, and z). This can disrupt the spatial relationships between points and potentially affect the embedded watermark.

b) Downsampling: This attack involves randomly removing points accounting for p of the total. It reduces the overall data density and can potentially damage the embedded watermark. In our setting, we randomly choose some points within the point cloud and set them to the mean of other points (0 after normalization).

c) Rotation: This attack involves randomly rotating the entire point cloud around the x, y, and z axes in angle θ. All of our used watermark extractors take *xyz* coordinates as input and rotation will change the value in spatial space, which will affect the output of networks directly. Rotation can significantly alter the spatial relationships within the point cloud and potentially disrupt the watermark.

3) Training: The goal of our watermark network is to effectively embed and extract the watermark under the premise of invisibly changing point clouds. There are two constraints that naturally should be taken into consideration: binary-entropy loss $\mathcal{L}_w$ for watermark and point-wise L2 loss $\mathcal{L}_{\text{recon}}$ between processed and original data. We found that only point loss leads to irregular shaking, so we introduced an additional variance regular item $\mathcal{L}_v$. The overall loss function is:

$$\mathcal{L} = \mathcal{L}_{\text{recon}}(p^w, p) + \beta_1 \mathcal{L}_w(\mathcal{W}_D(p^w), m) + \beta_2 \mathcal{L}_v(p^w, p) \tag{2}$$

while β_1 and β_2 are the weighted coefficients for the trade-off between accuracy and consistency.

B. Finetune Noise Predictor

The diffusion generation model is based on multiple denoising steps. In the early steps, the data are more like Gaussian noise, which can hardly be recognized as certain objects. So we divide it into two periods, the denoising period and the watermarking period. We define a demarcation point $\bar{t}$, which is determined by the robustness of the watermark extractor of Gaussian noise and the watermarking period ($t < \bar{t}$) aims to embed messages in synthesized data while the denoising period ($t \geq \bar{t}$) keeps the ability to generate point clouds.

To achieve these targets, we start with a pre-trained generative diffusion model, which learns the distribution of training data well. Then, we fine-tune the generative model using a watermark decoder mentioned in III-A to guide it to synthesize watermarked point clouds. Specifically, given a certain message m, we want to implant it into a diffusion model. We initialized two generators, G_{frozen} and G_{wm} with a pre-trained point clouds generative diffusion model.

$$\mathcal{L}_{\text{denoising}} = ||\epsilon_{\text{frozen}}(p_t, t) - \epsilon_{\text{wm}}(p_t, t)||_2^2, \tag{3}$$

$$\mathcal{L}_{\text{wm}} = \text{BCE}(\mathcal{W}_D(\hat{p}_{t-1}), m), t < \bar{t}, \tag{4}$$

where $\epsilon(\cdot, \cdot)$ is the noise predictor in diffusion models, $\hat{p}_t$ is the 1-step denoising results at time t and BCE is the binary cross-entropy loss. Thanks to our watermark decoder being robust to Gaussian noise, if the denoising diffusion models can generate watermarked point clouds at step t, the denoised point clouds should also be watermarked at step $t-1$. Finally, the watermark will be strengthened with several steps of denoising and watermarking.

IV. Experiments

A. Experimental Setup

1) Dataset: We employ airplanes from ShapeNet [38]. In both stages, we train/test our networks by following DPM [9]: sampling 2048 points from each point cloud, randomly splitting training, testing, and validation sets by the ratio 80%, 10%, and 5%, and most importantly, normalizing all the point clouds to zero mean and unit variance.

2) Baselines: Similar to DGCNN [4], there are a lot of backbones for extracting features from point clouds such as PointNet [1] and Point Cloud Transformer [3]. We employ them on our watermark extractor by replacing the global and local encoders and evaluate them by applying different attacks including Gaussian noise, rotation, and downsampling under the same control of $\mathcal{L}_{\text{recon}}$. The comparisons of different networks are shown in Table I. In our experiments, the DGCNN-based watermark extractor shows better robustness to different attacks, especially for Gaussian noise.

3) Implementation details: We employ all of our experiments on a single NVIDIA RTX3090. In the first stage for the watermark extractor, we first set $\beta_1 = \beta_2 = 1$ until convergence and set them to 200 subsequently. In the second stage, we finetune our watermarked diffusion models with $\lambda_i = \lambda_v = 1$ and demarcation point $\bar{t} = 5$. We adopt Adam as our optimizer with learning rates of 5×10^{-3}(first stage) and 5×10^{-4}(second stage). When root watermark in diffusion models, the watermark payload is 8 bits.

B. Overall Performance

Table II presents the generation quality and watermark accuracy of watermarked DPM [9], in terms of accuracy, Chamfer Distance (CD), and Earth Mover's Distance (EMD) [39]. Specifically, we finetuned the point cloud diffusion and sampled 1000 each of the watermarked airplanes, chairs, and tables as our validation sets. Notably, in the watermarking stage, we obtained the noisy point clouds by following the sampling process of DDPM [6] instead of DDIM [7]. The experimental results show better consistency and acceptable degradation in the case of DDIM(N=100, N=20), where N is the number of denoising steps. The starting point of the watermarking period $\bar{t}$ is set to 5, but sampling with DDIM(N=5) will step over the last five steps directly, leading to a drastic drop. Figure 3 visualizes the high consistency from the level of perception, which are sampled with the same random seeds and DDIM(N=20). We can find that the watermarked point clouds are very close to the original ones, which indicates that DPWD could effectively implant watermarks in the point clouds while well-preserving the quality.

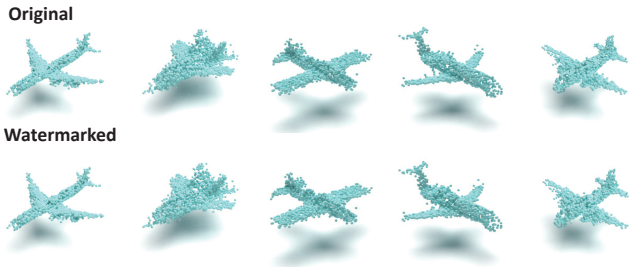

Fig. 3. Visual results of watermarked point clouds.

C. Simulated Attack Detection

This section delves into the resilience of watermarked diffusion models against adversarial challenges. In our study, we meticulously trained three specialized diffusion models, each dedicated to the generation of distinct object categories: airplanes, chairs, and tables. For each category, we generated a substantial dataset comprising 1000 point clouds to ensure a robust sample size. To rigorously assess the durability of our watermarked models, we subjected them to a diverse array of attacks, each varying in type and intensity. The resilience of the models was quantitatively measured, and the relationship between the severity of the attacks and the models' accuracy in watermark detection was carefully analyzed. The results of this analysis are graphically represented in Figure 4, which depicts the attack curves alongside the corresponding accuracy rates, providing a visual interpretation of the model's robustness under adversarial conditions.

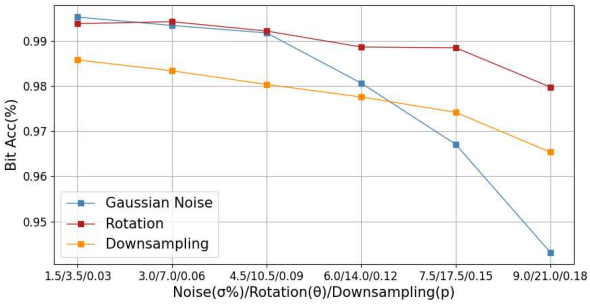

Fig. 4. Bit accuracy under different attacks.

D. Ablation Study

1) Modules in Watermark Extractor: Within the architecture of our watermark extractor, we have meticulously integrated three pivotal

TABLE I
COMPARSION OF DIFFERENT BASELINES

Payload	Backbone	Attacks										
		None	Gaussian Noise(σ)			Rotation(θ)			Downsampling(p)			Combination
		-	1%	3%	5%	5°	10°	20°	0.05	0.1	0.2	-
8-bits	PointNet	0.998	0.998	0.996	0.982	0.998	0.997	0.992	0.996	0.993	0.973	0.986
	PCT	0.998	0.998	0.997	0.994	0.998	0.996	0.981	**0.997**	0.995	0.973	**0.991**
	DPWD(ours)	0.998	0.998	**0.998**	**0.996**	0.998	0.997	**0.993**	0.996	0.995	**0.979**	**0.991**
16-bits	PointNet	0.995	0.994	0.983	0.947	0.995	0.993	0.983	0.993	0.991	**0.978**	0.980
	PCT	0.996	0.995	0.991	0.969	0.995	0.991	0.975	0.992	0.988	0.954	0.983
	DPWD(ours)	**0.997**	**0.997**	**0.995**	**0.984**	**0.998**	**0.997**	**0.988**	**0.995**	**0.992**	0.972	**0.986**
32-bits	PointNet	0.986	0.982	0.931	0.853	0.984	0.976	0.944	0.979	0.970	**0.944**	0.928
	PCT	0.994	0.992	0.979	0.916	**0.995**	0.986	0.959	0.987	**0.982**	0.909	**0.973**
	DPWD(ours)	**0.995**	**0.994**	**0.988**	**0.954**	0.994	**0.993**	**0.976**	**0.990**	0.980	0.917	0.967

The red: the best accuracy of different methods and attacks of the strongest ratio. The bold: the best accuracy of attacks of the other ratios unless there are multiple best ones. Combination: $\sigma = 1\%$, $\theta = 10°$, and $p = 0$ 1.

TABLE II
GENERATION QUALITY AND BITS ACCURACY

Sampler	Bit Acc(%)	CD(%)	EMD(%)
DDPM	98.9	0.0272	2.93
DDIM(N=100)	98.4	0.00773	1.09
DDIM(N=20)	98.1	0.00674	0.990
DDIM(N=10)	91.9	0.00678	1.01

modules, each designed to enhance the system's robustness against a spectrum of potential disruptions. To rigorously assess the individual contribution of these modules to the overall robustness, we embarked on a comprehensive training regimen for three additional modified extractors. Each of these extractors was deprived of a specific feature: the first lacked local features, the second was stripped of global features, and the third operated without the benefit of the attack layer.

To ensure an equitable comparison in our experiments, we compensated for the absence of the global or local encoder within the watermark extractor by proportionally increasing the number of channels. This adjustment was crucial to maintain the integrity of the study. The findings from this meticulous ablation study are systematically presented in Table III. The data therein clearly indicate that each module plays a significant role in fortifying the robustness of the watermark extractor. The results unequivocally demonstrate that the inclusion of local features, global features, and the attack layer each contribute substantially to the system's ability to withstand and adapt to various challenges, thereby validating the effectiveness of our modular design approach.

TABLE III
BITS ACCURACY OF ABLATION STUDY FOR MODULES IN PI-HIDDEN

Modules	None	Noise	Rotation	Downsampling
w/o f^l	0.898	0.875	0.856	0.883
w/o f^g	0.996	0.991	0.969	0.968
w/o attack layer	0.996	0.939	0.633	0.867
all	0.998	0.996	0.993	0.979

σ= 5%, θ = 20°, and p = 0.2.

2) Strategy of dual-period distilling: To assess the efficacy of our training methodology, we meticulously fine-tuned two distinct diffusion models, each operating under a different temporal framework: a single period and a dual period. The single-period model operates without taking into account the denoising process that occurs prior to time step $\bar{t}$. Conversely, our dual-period approach incorporates an additional distilling phase that significantly enhances the watermarking process, as depicted in Figure 2. This figure illustrates the comparative advantage of our dual-period distilling strategy over the conventional single-period approach, specifically within the watermarking phase.

The absence of denoising constraints in the earlier stage of the single-period model means that any fine-tuning applied to the noise predictor after time step $\bar{t}$ has the potential to inadvertently influence the denoising outcomes that precede $\bar{t}$. This inter-dependency introduces an additional layer of complexity to the training process, as the model must rely solely on the information gleaned from the final few steps. Our dual-period model addresses this challenge by integrating a pre-watermarking denoising phase, which stabilizes the training process and enhances the overall quality of the watermarking.

TABLE IV
ABLATION STUDY FOR THE STRATEGY OF DUAL-PERIOD DISTILLING

Setting	Bit Acc(%)	CD(%)	EMD(%)
single period	87.4	0.0294	3.70
dual periods	98.9	0.0272	2.93

V. CONCLUSION

In this work, we propose the Dual-Process Watermarked Diffusion (DPWD) framework, which marks a significant advancement in the protection of intellectual property for point cloud generation using Diffusion Models (DMs). Our innovative approach embeds watermarks directly into the noise predictor of DMs, ensuring the integrity and quality of the generated point clouds. The empirical studies validate the robustness of DPWD, demonstrating its effectiveness in embedding watermarks without compromising the functionality of the models or the quality of the generated point clouds. The resistance of the embedded watermarks to a variety of adversarial attacks establishes DPWD as a new benchmark for model protection.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[2] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[3] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "Pct: Point cloud transformer," *Computational Visual Media*, vol. 7, no. 2, p. 187–199, Apr 2021. [Online]. Available: http://dx.doi.org/10.1007/s41095-021-0229-5

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (tog)*, vol. 38, no. 5, pp. 1–12, 2019.

[5] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in neural information processing systems*, vol. 30, 2017.

[6] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Advances in neural information processing systems*, vol. 33, pp. 6840–6851, 2020.

[7] J. Song, C. Meng, and S. Ermon, "Denoising diffusion implicit models," *arXiv preprint arXiv:2010.02502*, 2020.

[8] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[9] S. Luo and W. Hu, "Diffusion probabilistic models for 3d point cloud generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2837–2845.

[10] A. Vahdat, F. Williams, Z. Gojcic, O. Litany, S. Fidler, K. Kreis *et al.*, "Lion: Latent point diffusion models for 3d shape generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 10 021–10 039, 2022.

[11] L. Zhou, Y. Du, and J. Wu, "3d shape generation and completion through point-voxel diffusion," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 5826–5835.

[12] S. Mo, E. Xie, R. Chu, L. Hong, M. Niessner, and Z. Li, "Dit-3d: Exploring plain diffusion transformers for 3d shape generation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[13] G. K. Nakayama, M. A. Uy, J. Huang, S.-M. Hu, K. Li, and L. Guibas, "Difffacto: Controllable part-based 3d point cloud generation with cross diffusion," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 14 257–14 267.

[14] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022.

[15] L. Wu, D. Wang, C. Gong, X. Liu, Y. Xiong, R. Ranjan, R. Krishnamoorthi, V. Chandra, and Q. Liu, "Fast point cloud generation with straight flows," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 9445–9454.

[16] P. Agarwal and B. Prabhakaran, "Robust blind watermarking of point-sampled geometry," *IEEE Transactions on Information Forensics and Security*, vol. 4, no. 1, pp. 36–48, 2009.

[17] A. Dong and R. Zeng, "Research and implementation based on three-dimensional model watermarking algorithm," in *2017 International Conference on Computing Intelligence and Information System (CIIS)*. IEEE, 2017, pp. 277–282.

[18] J. Liu, Y. Yang, D. Ma, Y. Wang, and Z. Pan, "A watermarking method for 3d models based on feature vertex localization," *IEEE Access*, vol. 6, pp. 56 122–56 134, 2018.

[19] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," in *Proceedings Shape Modeling Applications, 2004.* IEEE, 2004, pp. 233–242.

[20] F. A. Ferreira and J. B. Lima, "A robust 3d point cloud watermarking method based on the graph fourier transform," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 1921–1950, 2020.

[21] J. Liu, Y. Yang, D. Ma, Y. Wang, and Z. Pan, "A watermarking algorithm for 3d point cloud models using ring distribution," *Transactions on Edutainment XIV*, pp. 56–68, 2018.

[22] F. Xiaoqing, "A watermarking for 3d point cloud model using distance normalization modulation," in *2015 4th international conference on computer science and network technology (ICCSNT)*, vol. 1. IEEE, 2015, pp. 1449–1452.

[23] S. Zhang, F. Wang, and S. Zhai, "A novel watermarking algorithm for color point-cloud models based on 2d-dct," in *Advances in Natural Computation, Fuzzy Systems and Knowledge Discovery: Volume 2*. Springer, 2020, pp. 796–803.

[24] C. Zhang, P. Benz, A. Karjauv, G. Sun, and I. S. Kweon, "Udh: Universal deep hiding for steganography, watermarking, and light field messaging," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 223–10 234, 2020.

[25] J. Zhu, R. Kaplan, J. Johnson, and L. Fei-Fei, "Hidden: Hiding data with deep networks," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 657–672.

[26] P. Fernandez, A. Sablayrolles, T. Furon, H. Jégou, and M. Douze, "Watermarking images in self-supervised latent spaces," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022, pp. 3054–3058.

[27] J.-E. Lee, Y.-H. Seo, and D.-W. Kim, "Convolutional neural network-based digital image watermarking adaptive to the resolution of image and watermark," *Applied Sciences*, vol. 10, no. 19, p. 6854, 2020.

[28] Y. Wen, J. Kirchenbauer, J. Geiping, and T. Goldstein, "Tree-rings watermarks: Invisible fingerprints for diffusion images," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[29] R. Min, S. Li, H. Chen, and M. Cheng, "A watermark-conditioned diffusion model for ip protection," *arXiv preprint arXiv:2403.10893*, 2024.

[30] P. Fernandez, G. Couairon, H. Jégou, M. Douze, and T. Furon, "The stable signature: Rooting watermarks in latent diffusion models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 22 466–22 477.

[31] Y. Zhao, T. Pang, C. Du, X. Yang, N.-M. Cheung, and M. Lin, "A recipe for watermarking diffusion models," *arXiv preprint arXiv:2303.10137*, 2023.

[32] T. Qiao, Y. Ma, N. Zheng, H. Wu, Y. Chen, M. Xu, and X. Luo, "A novel model watermarking for protecting generative adversarial network," *Computers & Security*, vol. 127, p. 103102, 2023.

[33] J. Fei, Z. Xia, B. Tondi, and M. Barni, "Supervised gan watermarking for intellectual property protection," in *2022 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE, 2022, pp. 1–6.

[34] H. Wu, G. Liu, Y. Yao, and X. Zhang, "Watermarking neural networks with watermarked images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 7, pp. 2591–2601, 2020.

[35] C. Xiong, C. Qin, G. Feng, and X. Zhang, "Flexible and secure watermarking for latent diffusion model," in *Proceedings of the 31st ACM International Conference on Multimedia*, 2023, pp. 1668–1676.

[36] F. Wang, H. Zhou, H. Fang, W. Zhang, and N. Yu, "Deep 3d mesh watermarking with self-adaptive robustness," *Cybersecurity*, vol. 5, no. 1, p. 24, 2022.

[37] X. Zhu, G. Ye, X. Luo, and X. Wei, "Rethinking mesh watermark: Towards highly robust and adaptable deep 3d mesh watermarking," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 7, 2024, pp. 7784–7792.

[38] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su *et al.*, "Shapenet: An information-rich 3d model repository," *arXiv preprint arXiv:1512.03012*, 2015.

[39] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, pp. 99–121, 2000.