Crypto-Dropout: To Create Unique User-Generated Content Using Crypto Information in Metaverse

Haihan Duan
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
haihanduan@link.cuhk.edu.cn

Xiao Wu
White Matrix Inc.
Nanjing, China
wuxiao@whitematrix.io

Wei Cai*
School of Science and Engineering
The Chinese University of Hong Kong, Shenzhen
Shenzhen, China
caiwei@cuhk.edu.cn

***Abstract*—In a blockchain-driven metaverse, user-generated content (UGC) is the core power for building the metaverse, so an easy-to-use UGC editor is imperative. Specifically, using artificial intelligence (AI) to simplify the UGC creation procedure is promising, e.g., generating images from sketches using generative adversarial networks (GANs). However, the simplicity of these UGC creation methods would lead to weak distinctions between the generated UGC, since the users' created drafts may be very similar. In this paper, we propose Crypto-dropout, a specially designed dropout used in the generative neural networks, which could cause pseudo-random disturbance based on the hash value of user information to generate unique results. With a pilot study, the experimental results demonstrate that the participants have different preferences for the generated images when setting Crypto-dropout in the different layers. Accordingly, we implement a practical profile pictures (PFPs) creation prototype. The proposed Crypto-dropout can provide a novel and general insight for creating unique UGC using generative neural networks.**

***Index Terms*—User-Generated Content, Metaverse, Dropout, Human-centered Computing, Non-fungible Token**

I. Introduction

With the development of blockchain-related technologies, digital assets can be transformed to non-fungible tokens (NFTs) based on ERC-721 Standard[1] on Ethereum[2]. After that, the digital assets are given a public, unique, and non-interchangeable proof of ownership based on blockchain technology. Therefore, images, videos, 3D models, and other types of digital files can be stored as NFTs to confirm their ownership [1]. In the blockchain-driven metaverse, which will be the next generation social network, the digital assets that reflect the innovation, imagination, and creativity of users can truly belong to the users, which motivates more participants to join in the construction of the metaverse [2], [3]. As a promising trend, user-generated content (UGC) will play a necessary role in the metaverse [4].

To this end, most metaverse projects provide UGC editors for their users, e.g., *Cryptovoxels*[3], *Decentraland*[4], *The Sandbox*[5], etc. However, most normal users of the metaverse are not experts in drawing or 3D modeling, which influences their confidence and enthusiasm to participate in the building of metaverse, so the metaverse-equipped editors should be easy-to-use for amateurs. With the development of artificial intelligence (AI), especially deep learning [5], many generative models (like variational auto-encoder (VAE) [6] and generative adversarial network (GAN) [7]) showed very impressive and powerful performance, e.g., pix2pix [8], CycleGAN [9], GauGAN [10], SketchyGAN [11], etc., which could be utilized to assist the UGC creation of metaverse users [4].

In our prototype *The Chinese University of Hong Kong, Shenzhen (CUHKSZ) Metaverse* [2], we provide an AI-assisted editor to help users in creation profile pictures (PFPs). Using this editor, the users only need to draw some simple sketches, and the editor could generate a colorful PFP based on pix2pix [8], which effectively reduces the threshold of PFP creation. In fact, the PFPs are the most straightforward and intuitive reflection of users' personal lifestyle, community, and experience, so the users always think highly of the uniqueness of their PFPs [12], [13]. However, the simplicity of the AI-assisted UGC creation method would lead to weak distinctions between the generated UGC. For example, many amateurs may use a stroke to draw the face, a stroke to be the nose, a stroke to denote the mouse, and two circles to represent the eyes, etc., and the layout of these sketches would be very similar, which makes the generated PFPs are hard to distinguish and lack of uniqueness and personality. Therefore, it is imperative to design a mechanism that could guarantee the uniqueness of the UGC generated by AI-assisted editors.

In the crypto community, the hash function can convert any digital information to a pseudo-random code with a fixed length [14], a key technique in the proof of work (PoW) consensus model [15], so-called crypto information. Motivated by the feature of the hash function, we consider the crypto information of users can be utilized to control generative models individually. Therefore, we counter-intuitively apply dropout [16] in the generation of the generative neural network, named Crypto-dropout, which is controlled by the crypto information to provide pseudo-random disturbance. Compared with pure random disturbance, the Crypto-dropout can easily reproduce the generated results based on the identical user information and generative model. Specifically, this is a general idea that

*Wei Cai is the corresponding author (caiwei@cuhk.edu.cn).

[1]https://eips.ethereum.org/EIPS/eip-721
[2]https://ethereum.org/en/
[3]https://www.cryptovoxels.com/
[4]https://decentraland.org/
[5]https://www.sandbox.game/en/

can be applied in most neural networks.

After designing the Crypto-dropout, we conduct a pilot study with 11 participants to evaluate the human sense of generated results. The participants are asked to raise their preference when showing multiple image pairs generated by setting Crypto-dropout in different neural network layers, which can provide some insights for the system design of related studies. Moreover, we design and implement a practical PFP creation prototype based on the Crypto-dropout in CUHKSZ Metaverse, which can help users to generate unique anime PFPs using only simple sketches.

Our contributions can be concluded as follows:

- We propose a specially designed dropout technique used in the neural networks, named Crypto-dropout, for causing pseudo-random disturbance based on the hash value of user information to generate unique UGC.
- We conduct a pilot study on DCGAN [17] to evaluate the human sense of the Crypto-dropout. The experimental results can reflect the participants' preference for setting Crypto-dropout in different layers.
- We design and implement a PFP editor based on Crypto-dropout, which could generate unique PFPs for different users based on their crypto information and sketches.

II. Related Work

A. Non-Fungible Token-Based Content Creation

Based on ERC-721 Standard, NFT-based user-generated content (UGC) has attracted enormous industry attention, and lots of UGC platforms are springing up. For example, Art Blocks[6] is a platform focused on programmable generative content that is stored immutably on Ethereum. Moreover, Artnome[7] and Kate Vass Gallery[8] are similar to the galleries in the physical world, where artists can communicate their generative digital art. In these platforms, many artworks are generated using generative HTML Canvas script, like p5js. However, using scripts/codes to generate UGC in metaverse has higher barriers for normal users due to the basic requirement of programming, while applying generative models to generate content requires fewer fundamentals, so there are many notable projects created by generative models that have been known to the public. For example, Gan Chans[9] is a collection of 256 manual selected anime avatar NFTs generated by GAN. On Artnome, there are also many artworks created by generative models, like style transfer auto-encoder and GAN. In Omniverse[3], a metaverse for 3D design collaboration, NVIDIA also introduces GAN to transfer photos to 3D models. However, if using the same generative model and input, different users might generate identical results. Therefore, in this paper, we intend to apply crypto information of users to make the generated results unique and keep digital scarcity.

[6]https://www.artblocks.io/project/207
[7]https://www.artnome.com/news/2018/3/29/ai-art-just-got-awesome
[8]https://www.katevassgalerie.com/
[9]https://nftganchan.github.io/home/

B. Crypto Thinking in Neural Networks

In the deep learning area, most works considering crypto information are to solve problems about security and privacy. For example, Xie *et al.* [18] proposed Crypto-nets to promise the privacy requirement when a third party uses private user information in neural network prediction, which applied homomorphic encryption and modifications to the activation functions and training algorithms of neural networks. Hajjaji *et al.* [19] proposed a medical image crypto-compression algorithm based on the neural network and the chaotic system. Shafran *et al.* [20] optimized the design of crypto-oriented neural architectures to address the privacy issue of sending private data to neural network applications. Nandakumar *et al.* [21] evaluates the feasibility of training a neural network in a non-interactive way on data after fully homomorphic encryption. Our proposed Crypto-dropout shows a totally different perspective compared with the fore-mentioned works. We do not pay attention to security and privacy but apply the crypto information in the generation process of the generative neural network models to cause the pseudo-random disturbance of the generated results.

III. Methodology

A. Background of Dropout

Dropout is a technique proposed by G. E. Hinton *et al.* [16], which is utilized in fully-connected layers to prevent the overfitting problem during the training process of neural network models. During the model training, the dropout technique can randomly select some neurons and remove the neurons, which means the weights connected to the dropout neurons will not affect the forward propagation and will not be updated. So, applying dropout to a neural network intends to sample a "thinned" neural network during the training process. After that, the complete neural network without dropout will be utilized in the prediction process.

B. Crypto-Switches Generation

In this paper, our motivation is to use a hash function to encrypt the user information as switches (so-called Crypto-switches) to control the dropout of neurons. We first select a layer as the Crypto-dropout layer and then number the neurons of this layer. According to the feature of the hash function, input with any length will output a hash value with a fixed size, so the users can customize the information for generating Crypto-switches. Supposing there are n neurons and the applied hash function is SHA-256 [22], the Crypto-switches can be generated as shown in Algorithm 1. The algorithm will generate a number that has n bits, and each bit matches a neuron of the Crypto-dropout layer in sequence. In general, the input information can be any length or any type of digital content that can be encoded by a hash function. The most important feature of this methodology is that, using the Crypto-dropout, the user information can pseudo-randomly impact the generated results rather than a fixed one or a purely random one that cannot be controlled and reproduced. On the other hand, theoretically, the difficulty of generating identical

Algorithm 1: Crypto-switches Generation

Input: Number of Neurons n, User Information I
Output: Crypto-switches S

```
Init: Crypto-switches S = 0
for i = 0 to i = n//256 + 1 do
    if n <= 256 then
        S = S << n;
        temp = SHA-256(Timestamp, I, ...) ;
        S = S OR (temp >> (256 − n))
    end
    else
        S = S << 256;
        S = S OR SHA-256(Timestamp, I, ...);
    end
    n = n − 256;
end
return S;
```

results is almost equal to the possibility of hash collision, which could guarantee the uniqueness of the generated results. In fact, due to the characteristics of neural networks, there are some neurons that have less influence on the output [23] (also named redundant neurons), so removing these neurons may not effectively impact the generated results, which will be discussed in Sec. VI.

C. Crypto-Dropout in Neural Networks

Due to the length limit, we will only discuss the most commonly used layers of generative neural network models, including the fully-connected and convolutional layers.

1) Crypto-Dropout in Fully-connected Layer: In deep learning, Fully-connected layer denotes the neuron has full connections with every neuron in the previous and subsequent layers, which is a basic structure of most neural network models. An example of Crypto-dropout in a fully-connected layer is shown in Figure 1, where a three-layer basic neural network model is illustrated, which has 4 neurons in each fully-connected layer. In this example, we select the middle layer as the Crypto-dropout layer and number the neurons from 0 to 3. Then we calculate the hash value as Crypto-switches according to the user information, where we assume the Crypto-switches is 0101, which corresponds to the $0th$ - $3rd$ neurons. As a result, the $0th$ and $2nd$ neurons will be removed in the generation.

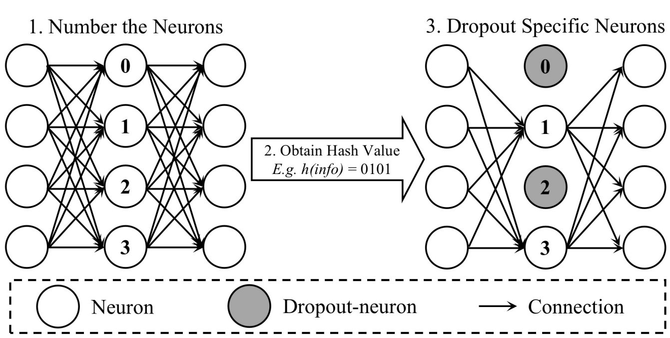

Fig. 1. Illustration of Crypto-dropout in fully-connected layer. (1) Number the neurons of the Crypto-dropout layer; (2) Calculate the hash value using provided information (e.g. 0101); (3) Dropout the corresponding neurons according to the hash value (0: dropout, 1:retain).

2) Crypto-Dropout in Convolutional Layer: The convolutional layer applies a convolution operation to the input to obtain the information from a 2-dimensional view, typically a dot product of the convolution kernel with the input matrix. Convolutional neural network (CNN) has achieved great performance in computer vision tasks [5], and the content generation tasks are also benefited a lot. The study about dropout in a convolutional layer was deeply discussed by previous works [24], [25], which mentioned various methods of dropout in a convolutional layer, including drop-neuron, drop-channel, drop-path, and so on. In this paper, we only illustrate the drop-channel method, as shown in Figure 2, due to the length limit. In this example, Figure 2 illustrates a kernel of a convolutional layer, which totally has 8 channels and is selected as the Crypto-dropout layer, and the channels are numbered from 0 to 7. Then we assume the Crypto-switches is 10011010 obtained by a hash function, which means the $1st$, $2nd$, $5th$, and $7th$ channel in the kernel of the convolutional layer will be removed in the generation process of the neural network model. In general, the same methodology could also adapt the drop-neuron, drop-path, etc., during the practice.

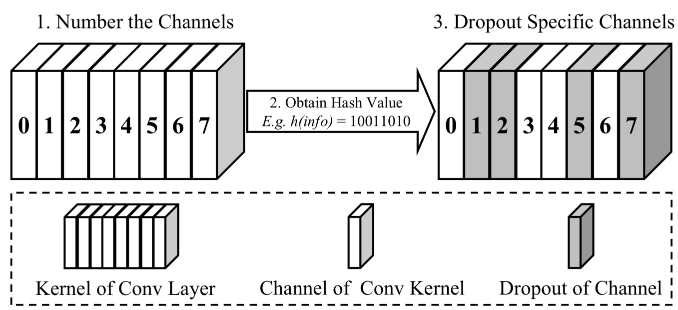

Fig. 2. Illustration of Crypto-dropout in convolutional layer. We use drop-channel as an example. (1) Number the channels of convolutional kernel of the Crypto-dropout layer; (2) Calculate the hash value using provided information (e.g. 10011010); (3) Dropout the corresponding channels according to the hash value (0: dropout, 1:retain).

IV. Pilot Study

A. Experimental Settings

To evaluate the performance of the proposed Crypto-dropout, we conduct experiments on the classical generative neural network model DCGAN [17]. In the generation, we use different Ethereum addresses of the authors as the user information to generate multiple image pairs. During the generation, we maintain the neural network model, input parameter, and timestamp not change while only changing the position for setting the Crypto-dropout layer. As expected, the generated image pairs should be different, as discussed in Sec. III-B. In fact, the difference can be simply proved by numerical comparison using mean square error (MSE). However, we intend to study the users' perception of different Crypto-dropout settings from a human-centered perspective. Therefore, we conduct a pilot user study to invite the participants to select the results of different Crypto-dropout layers that they consider to have a better visual experience. The experimental results can be a reference in the design of practical applications.

B. Generative Neural Network Implementation

In most novel GANs, the convolutional layer plays a significant role, so we want to study the effectiveness of Crypto-dropout in a convolutional layer. We construct a simple DCGAN model [17], as shown in Figure 3, which includes both discriminator and generator. In this experiment, we will apply Crypto-dropout in the generator, and the dropout method is drop-channel, as shown in Figure 2. Note that, in this experiment, the Crypto-dropout is actually applied on a transposed convolutional layer rather than the standard convolutional layer, which also illustrates the generalization ability of the proposed Crypto-dropout. In the generation, we set Crypto-dropout in ConvTr1, ConvTr2, and ConvTr3 to compare the difference between the generated images.

The training dataset we utilized is CIFAR-10 [26], which consists of 32×32 colored scenery images. The training and testing set of CIFAR-10 contain 50,000 and 10,000 images, which are classified into 10 classes. The neural network model is implemented by PyTorch [27] and trained on NVIDIA RTX 3090 GPU. Adam [28] is used as the optimizer, and the learning rate is selected as 6×10^{-6} with beta of 0.5 and 0.999. Totally 100 epochs are conducted in training.

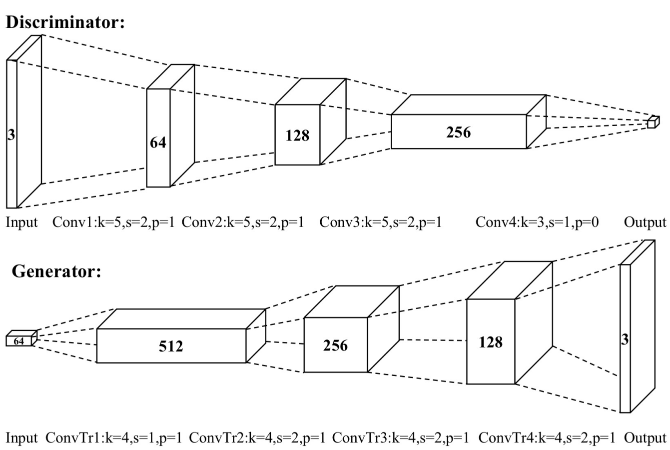

Fig. 3. The experimental structure of a simple DCGAN, including both discriminator and generator. In this figure, "Conv" denotes the 2D convolutional layer and "ConvTr" means the 2D transposed convolutional layer, with "k" as kernel size, "s" as stride, and "p" as padding. And the channel numbers are marked on each layer.

C. Experimental Results

We randomly select and keep a set of input parameters to generate 40 pairs of images by setting Crypto-dropout on ConvTr1, ConvTr2, ConvTr3, and ConvTr4 for DCGAN, a total of 40 images. Some sample results are shown in Figure

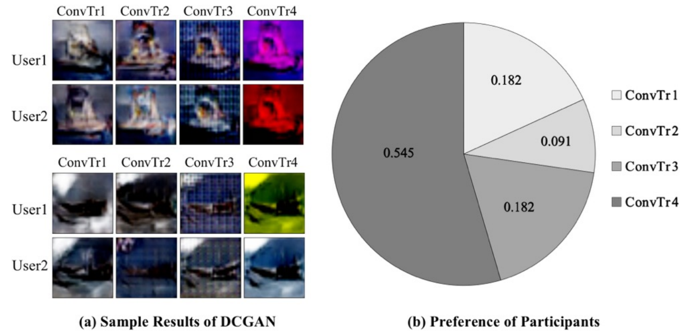

Fig. 4. Statistical results of user study. The subfigure (a) is the experimental results of VAE, and the subfigure (b) is DCGAN. In these figures, the marked numbers are the average proportion of the selections by all participants.

4, where each block denotes generated results using the same neural network, input parameter, and timestamp, but different user information. Note that, the MSE values of all image pairs are higher than 0, which guarantees their numerical difference.

The sample results in Figure 4 illustrate the same conclusion, where the images of setting Crypto-dropout in ConvTr1 have apparent differences in pixel distribution while the images of ConvTr4 seem to be different only in color selection. This result is in accord with the characteristics of convolutional neural networks that features in the higher layers control the semantic information, and the lower layers adjust the texture. Regarding the user preference, most of them selected ConvTr4, which is close to the output of DCGAN. We also conducted a brief discussion about their preference. Most participants prefer strong color contrast because, compared with the dissimilar structure generated by ConvTr1, the different color distribution can give them a more intuitive perception of dissimilarity, which inspires us to pay more attention to the color distribution when producing unique PFPs in practice.

V. Profile Picture Creation Prototype

In this paper, we design a web-based user-friendly PFP creation prototype as a demo application of the proposed Crypto-dropout, which is to generate anime PFP from simple sketches driven by GAN. The graphic user interface (GUI) of the prototype is shown in Figure 6, named Crypto Profile Picture Canvas. In this demo, the left side canvas allows the users to draw sketches arbitrarily, and there is a toolbar upon the canvas that provides multiple helpful functions like do, undo, shapes, eraser, bucket, etc., which is modified from an open-source jQuery paint plugin wPaint.js[10]. Users can link their Ethereum wallet (e.g., MetaMask[11]) to connect with our prototype, and their Ethereum address will be recorded as input information. After drawing and providing the information, the users can click the "Generate Profile Picture" button, and an anime PFP will appear on the right side canvas, which is generated by a GAN model with Crypto-dropout based on the provided Ethereum address and the timestamp.

[10]https://github.com/websanova/wPaint
[11]https://metamask.io/

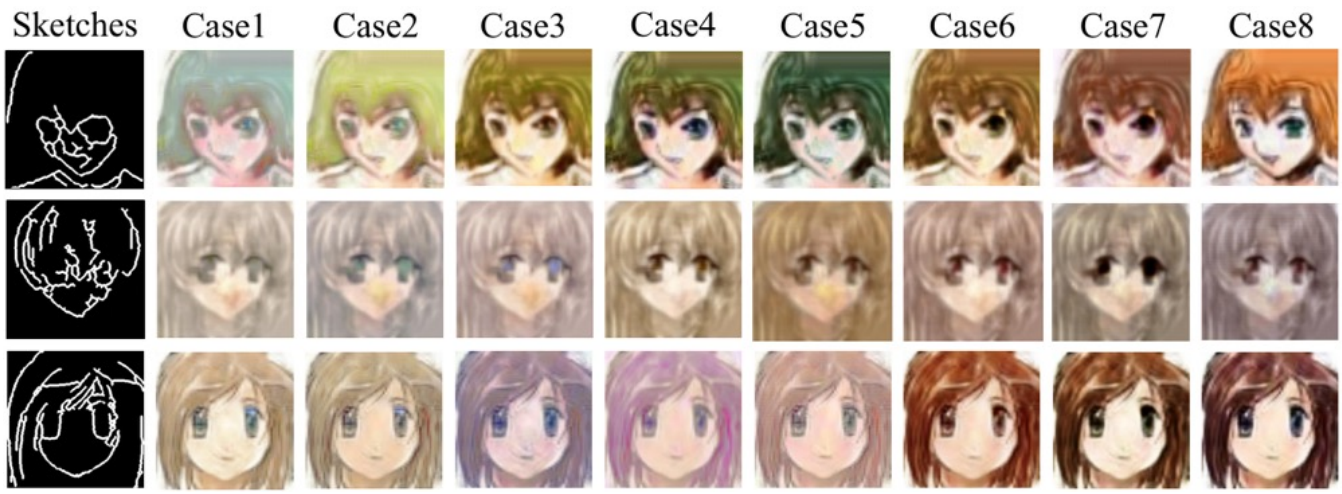

Fig. 5. Some Sample Images Generated from the Same Sketches by Inputting Different User Information

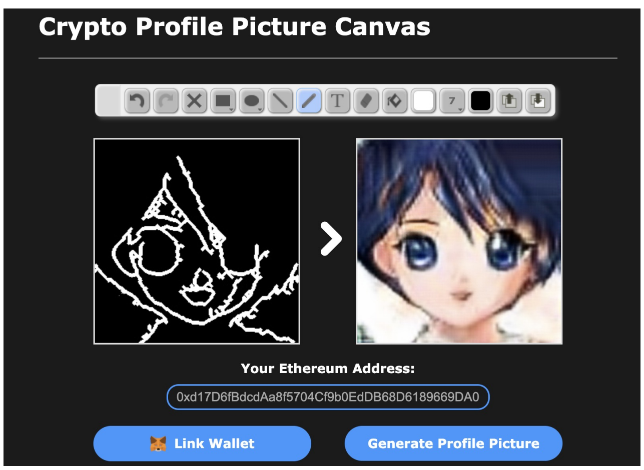

Fig. 6. Crypto Profile Picture Canvas Prototype

The GAN model we applied is pix2pix [8], a classic image-to-image translation method with an input and output resolution of 256×256. An open-source anime face dataset from Kaggle[12] is used for the model training, which contains 63,632 anime faces. Firstly, all images are resized to 256×256. The edges are detected using Holistically-nested edge detection (HED) [29], and then the OTSU adaptive binarization algorithm [30] is applied to the output images. To clean the redundant edges and enhance the sketches, we apply morphology thin, removing small objects, and dilation using scikit-image [31]. The model training utilizes the same experimental platform in Sec. IV-B. We used the Adam [28] optimizer, and the learning rate is 2×10^{-4} with the beta of 0.5 and 0.999, and it cost 200 epochs for model training.

According to the experimental results of the pilot study discussed in Sec. IV-C, we select the convolutional layer closing to the output (the "OutConv" layer in pix2pix [8]) as the Crypto-dropout layer. Figure 5 shows some sample images generated from the same sketches by different timestamps. We can find that, although the generated results using different crypto information have similar structures, their color distributions show significant differences, which is in accord with the conclusions of the pilot study as discussed in Sec. IV-C.

VI. Limitations and Future Work

According to the feature of the neural network, the functions of neurons are highly different. Some neurons have less influence on the output (so-called redundancy of neural networks), while some neurons store the critical information for constructing meaningful results [23]. Although the Crypto-switches have base information, the generated Crypto-switches could be regarded as pseudo-random due to the characteristics of the hash function, so the randomness makes the generated content hard to be controlled. On the one hand, if the dropped layers contain vital information for semantic construction, the model might generate weird results. On the other hand, if the dropped layers are redundant, the generated results may not present noticeable visual differences.

Therefore, the proposed Crypto-dropout may not be the best solution. However, the motivation of this paper is to "throw a sprat to catch a whale" and attract more attention to the uniqueness and digital scarcity of UGC in the metaverse. In the future, the weakness of the current Crypto-dropout needs to be addressed. For example, there are many researchers who study neural network pruning [32]–[34] and compression [23], [35]–[37], where the retained neurons after compression usually show a significant impact on the generated results. Thus, a possible solution is to combine neural network compression and feature visualization [38] to select neurons that control the artistic style rather than the semantics of images. Last but not least, the results shown in Figure 5 indeed lose some details of the anime faces, which means the direct dropout may cause a loss of information. Therefore, methods like the average of channel values can be utilized to replace the direct

[12] https://www.kaggle.com/splcher/animefacedataset

dropout of the neurons or layers, which might be smoother when impacting the generated results.

VII. Conclusion

This paper proposes Crypto-dropout, a specially designed dropout that can take user information into account using a hash function to cause pseudo-random in UGC creation. We conduct a pilot study to evaluate the human reaction to the generated results of Crypto-dropout, then we design and implement a practical prototype to create unique PFPs according to the experimental results. We believe this idea is a novel and insightful solution for keeping the digital scarcity of UGC creation using generative neural networks in the metaverse, which has considerable potential for further improvement and wider application scenarios, e.g., inspiring the editor design of 3D modeling and digital twin.

Acknowledgment

This work was supported by Shenzhen Science and Technology Program (Grant No. JCYJ20210324124205016).

References

[1] Q. Wang, R. Li, Q. Wang, and S. Chen, "Non-fungible token (nft): Overview, evaluation, opportunities and challenges," *arXiv preprint arXiv:2105.07447*, 2021.

[2] H. Duan, J. Li, S. Fan, Z. Lin, X. Wu, and W. Cai, "Metaverse for social good: A university campus prototype," in *Proceedings of the 29th ACM International Conference on Multimedia*, 2021, pp. 153–161.

[3] L.-H. Lee, T. Braud, P. Zhou, L. Wang, D. Xu, Z. Lin, A. Kumar, C. Bermejo, and P. Hui, "All one needs to know about metaverse: A complete survey on technological singularity, virtual ecosystem, and research agenda," *arXiv preprint arXiv:2110.05352*, 2021.

[4] H. Duan, Y. Huang, Y. Zhao, Z. Huang, and W. Cai, "User-generated content and editors in video games: Survey and vision," in *2022 IEEE conference on games (CoG)*, 2022, pp. 1–8.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[6] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[7] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125–1134.

[9] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2223–2232.

[10] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2337–2346.

[11] W. Chen and J. Hays, "Sketchygan: Towards diverse and realistic sketch to image synthesis," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9416–9425.

[12] J. Chang, H. Ren, and Q. Yang, "A virtual gender asylum? the social media profile picture, young chinese women's self-empowerment, and the emergence of a chinese digital feminism," *International Journal of Cultural Studies*, vol. 21, no. 3, pp. 325–340, 2018.

[13] M. A. Baytaş, A. Cappellaro, and Y. Fernaeus, "Stakeholders and value in the nft ecosystem: Towards a multi-disciplinary understanding of the nft phenomenon," in *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, 2022, pp. 1–8.

[14] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 769–790, 2017.

[15] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, p. 21260, 2008.

[16] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.

[17] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

[18] P. Xie, M. Bilenko, T. Finley, R. Gilad-Bachrach, K. Lauter, and M. Naehrig, "Crypto-nets: Neural networks over encrypted data," *arXiv preprint arXiv:1412.6181*, 2014.

[19] M. A. Hajjaji, M. Dridi, and A. Mtibaa, "A medical image crypto-compression algorithm based on neural network and pwlcm," *Multimedia Tools and Applications*, vol. 78, no. 11, pp. 14 379–14 396, 2019.

[20] A. Shafran, G. Segev, S. Peleg, and Y. Hoshen, "Crypto-oriented neural architecture design," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 2680–2684.

[21] K. Nandakumar, N. Ratha, S. Pankanti, and S. Halevi, "Towards deep neural network training on encrypted data," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019.

[22] N. I. of Standards and Technology, "Announcing approval of federal information processing standard (fips) 180-2, secure hash standard; a revision of fips 180-1," 2003.

[23] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[24] H. Wu and X. Gu, "Towards dropout training for convolutional neural networks," *Neural Networks*, vol. 71, pp. 1–10, 2015.

[25] S. Cai, Y. Shu, G. Chen, B. C. Ooi, W. Wang, and M. Zhang, "Effective and efficient dropout for deep convolutional neural networks," *arXiv preprint arXiv:1904.03392*, 2019.

[26] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in neural information processing systems*, 2019, pp. 8026–8037.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[30] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[31] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu, "scikit-image: image processing in python," *PeerJ*, vol. 2, p. e453, 2014.

[32] D. Blalock, J. J. Gonzalez Ortiz, J. Frankle, and J. Guttag, "What is the state of neural network pruning?" *Proceedings of machine learning and systems*, vol. 2, pp. 129–146, 2020.

[33] P. Molchanov, A. Mallya, S. Tyree, I. Frosio, and J. Kautz, "Importance estimation for neural network pruning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 264–11 272.

[34] Z. Wang, C. Li, and X. Wang, "Convolutional neural network pruning with structural redundancy reduction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 913–14 922.

[35] J.-H. Luo, J. Wu, and W. Lin, "Thinet: A filter level pruning method for deep neural network compression," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5058–5066.

[36] H. Kim, M. U. K. Khan, and C.-M. Kyung, "Efficient neural network compression," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 569–12 577.

[37] Y. Li, S. Gu, L. V. Gool, and R. Timofte, "Learning filter basis for convolutional neural network compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5623–5632.

[38] C. Olah, A. Mordvintsev, and L. Schubert, "Feature visualization," *Distill*, vol. 2, no. 11, p. e7, 2017.