Supplementary Material

1 Rethinking Dataset Copyright Ownership in the Era of Generative AI

Due to the explosive growth of generative AI, an increasing number of creators' works, including creative entities, brushstrokes, and styles, are being used for unauthorized profit. In Glaze[1]'s survey, based on responses from 1,207 artists, the vast majority hope for fair legislation to protect the unique artistic styles and content of their works. However, there are currently no feasible solutions, and this issue is highly challenging. Currently, generative AI is being maliciously used by some to easily learn, imitate, and plagiarize unauthorized human works for profit. This severely undermines creators' motivation, damages their creative enthusiasm, and turns high-quality works into others' benefits. This step may be urgent for ensuring intellectual property rights for human creativity in the AI era. Therefore, we aim to establish a positive and healthy cycle for the development of art between generative AI and human creation.

2 The Relevant Explanation for Identifier z

In this paper, identifier z is designed to ensure the exclusivity and uniqueness of the protected unit distribution within large datasets. Identifier z signifies the identifier that maximally shifts the contraction domain to the edge distribution of the style representation space. In other word, the domain achieves the maximum concealed offset of probability distribution through both the injection of identifier z, after decoupling the style domain and performing dynamic contrastive learning to increase the distance in the similarity space. Since is an arbitrary identifier (such as text, strings, images, etc.), its capacity is effectively infinite, which further enhances the reliability and security of the solution. In machine learning, there are inherent differences in the high-dimensional feature distributions of protected units. Our approach, which utilizes the identifier, decouples the style domain and employs dynamic contrastive learning, aims to shift this distribution.

3 Method

3.1 The Style Domain Encoder and Decoder in Diffusion

The Style domain encoder and decoder are formally defined by a pair of forward and backward Markov chains representing a T-steps transformation from a normal distribution $z_T \sim \mathcal{N}(0, 1)$ into the learned distribution $z_0 \sim p_\theta(z_x)$. Each forward step t erodes x_t by adding a small Gaussian noise according to a fixed variance schedule α_t, sampling:

$$z_T \sim \mathcal{N}(\sqrt{\alpha_t}z_t - 1, \sqrt{1 - \alpha_t}I). \tag{1}$$

Meanwhile, each reverse step t performs image denoising, and aims to estimate ϵ_t in order to recover

$$p_\theta(z_{t-1}|z_t, s, c), \tag{2}$$

where the style representation $s \in \mathbf{R}^{\mathcal{D}}$ serves as a guidance source for denoising the image, and c denotes optional conditions for the encoder and decoder respectively. The reverse step is realized by a denoising decoder D_z that predicts:

$$z_T \sim \mathcal{N}(\sqrt{\alpha_t}z_t - 1, \sqrt{1 - \alpha_t}I). \tag{3}$$

Meanwhile, each reverse step t performs image denoising, and aims to estimate ϵ_t in order to recover

$$\epsilon = \epsilon_\theta(z_t, t, s, c). \tag{4}$$

Thanks to the reparametrization trick, we can then sample the following:

$$z_T \sim \mathcal{N}(\sqrt{\alpha_t}z_t - 1, \sqrt{1 - \alpha_t}I). \tag{5}$$

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Meanwhile, each reverse step t performs image denoising, and aims to estimate ϵ_t in order to recover

$$z_{t-1} \sim \mathcal{N}(\frac{1}{\sqrt{\alpha_t}}(z_t - \frac{1 - \alpha_t}{\sqrt{1 - \bar{\alpha}_t}}\epsilon_\theta(z_t, t, s, c)), \sigma_t^2 I), \tag{6}$$

where $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_s$ s the product of the variances up to step t, and σ_t^2 is either a fixed or learned variance term.

3.2 Self-Generalization by dynamic contrastive learning

The self-generalization of the contraction domain of protected samples is essentially the result of soft boundary delineation of the style domain. Its goal is to maximize generalizability, meaning that, aside from a series of versions of the proxy model, infringement models generated by other different architectures can also be effectively traced back.

Initialize Cetral and Boundary Samples. For a given data point x, the latent encoder (i.e., VAE) outputs the parameters of the probability distribution $q_\phi(z_x|x)$ for the latent z_x, which is the mean μ and variance σ^2 of z_x:

$$q_\phi(z_x|x) = \mathcal{N}(z_x; \mu(x), \sigma^2(x)I) \tag{7}$$

Here, ϕ denotes the parameters of VAE, and $\mu(x)$ and $\sigma^2(x)$ are calculated by VAE based on the input x. The z_x implies that x is regularized into a Gaussian distribution $\mathcal{N}$. Let $O = \{z_o\}_{o=1}^{K}$ which denotes the latent of protected units, where z_o shape is [B, 4, 64, 64] (i.e., B is the number of images of the o-th protected unit). And let $G_o = \{z_{g_o}\}_{g=1}^{N}$ denotes the mimic samples of protected units, where z_o shape is [B, 4, 64, 64] . We first normalize all latents as Eq.9 and Eq.6,

$$z_o = \frac{\mathcal{F}_{abs}(z_o)}{\|z_o\|_2}, \tag{8}$$

$$z_{g_o} = \frac{\mathcal{F}_{abs}(z_{g_o})}{\|z_{g_o}\|_2}, \tag{9}$$

where $\mathcal{F}_{abs}$ denotes the absolute value function. Next, We calculate the average difference between O and G_o.

$$e = \frac{1}{N} \sum_{z_{g_o} \sim \mathcal{G}_o} \sqrt{|z_o - z_{g_o}|_2}, \tag{10}$$

where e denotes the Threshold of G_o. If $z_{g_o} - e > 0$, it indicates a strong positive correlation; otherwise, it is a weak positive correlation. Similarly, we also performed the same operation at the image level. Finally, we compute the cosine similarity between the O and G_o in Eq.11.

$$sim = cos(\mathcal{E}_z(z_o), \mathcal{E}_z(z_{g_o})), \tag{11}$$

where $\mathcal{E}_z$ denotes the style domain encoder. If the sim is greater than χ (In our setting, $\chi = 0.9975$), it is marked as a strong positive correlation; otherwise, it is a weak positive correlation. We consider all samples that satisfy the strong positive correlation condition as central samples for the k-th unit and those that do not satisfy it as marginal samples. It is worth mentioning that, as the style domain encoder is updated, we dynamically update the distribution of central and marginal samples based on the cosine similarity threshold. The advantage of this approach is that it allows for decoupling the representation based on generalization when constructing the style domain. The dynamic central and marginal samples enable self-generalization of the style domain through dynamic contrastive learning, facilitating the structured delineation of dataset copyright boundaries for multiple sources of styles and contents in image generation.

4 Implament Details

4.1 Model Details

The configuration file outlines the specifications for training a neural network model, encompassing both the encoder and decoder components and parameters related to the diffusion process. On one hand, The encoder employs a ResNet-18 architecture with a feature dimension set to 128. On the

2

other hand, the decoder is configured with specific settings such as the shape of the output latent set to [4, 64, 64], the number of channels at 128, and the number of blocks set to 2. It also includes parameters like channel multipliers attention mechanisms, dropout rate, and whether to use residual connections for upsampling and downsampling. The dimensionality of the latent space is specified as 128 ($z_{channels}$). Regarding the diffusion process, the configuration sets the beta values to [1.0e-4, 0.02], indicating the scale of the diffusion process. The number of diffusion steps is set to 1000, determining the duration of the diffusion process. A dropout probability of 0.1 is also specified.

4.2 Experiment Details

The configuration defines various parameters for training the neural network model. It sets the batch size to 128, indicating the number of training examples processed in each iteration. The learning rate is specified as 1.0e-4 with a learning rate ratio of 2, suggesting a doubling of the learning rate over time. The 'AdamW' optimizer is chosen for optimization, with weight decay set to 0.05 and beta values of [0.9, 0.95]. Gradient clipping is applied with a maximum norm of 1 to prevent exploding gradients. The training process consists of 70 epochs, with a warm-up period of 20 epochs. The model is initialized either from scratch or from a pre-trained checkpoint, depending on the value of the load epoch parameter (-1 for training from scratch). Additionally, an exponential moving average (EMA) factor of 0.9999 is applied to stabilize the training process. These parameters collectively facilitate effective training of the neural network model. The paper involves the following parameters: a momentum coefficient $m = 0.999$, a temperature coefficient $\tau = 0.07$, a boundary constant $c = 1.0$, an exclusivity beta $\beta = 0.1$, 50 protected units K, 100 generalized samples N, and a watermark length $L = 128$ bits. In addition, We conducted all experiments for the paper on four Nvidia RTX 3090 GPUs. The disentangled pre-training took approximately 400 GPU hours on two NVIDIA GeForce RTX 3090s. We conducted over 100 sets of experiments on five benchmark datasets, with each set of experiments taking about 5 GPU hours per unit.
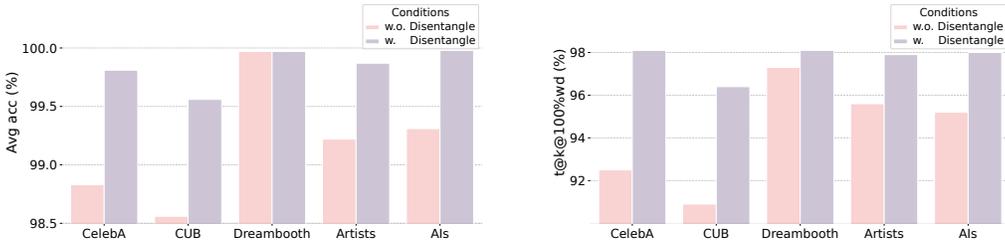
Figure 1: Compare the performance of our method in terms of $Avg\ acc$ and $t@k@100\%wd$ across five datasets, evaluating the impact of Disentanglement with and without ($w.$ and $w.o$)

Table 1: Main results. We evaluated the performance of the model's discriminator using the metrics of Accuracy, Precision, Recall, and F1-Score.

Dataset	Accuracy (%)	Precision (%)	Recall (%)	F1 (%)	$Avg\ acc$(%)	$t@k@100\%wd$ (%)
CelebA	99.79	99.72	100	99.87	**99.81**	**98.1**
CUB	99.21	99.86	100	99.93	**99.56**	**96.4**
Dreambooth	99.60	99.71	99.69	99.70	**99.97**	**98.1**
Artists	99.35	99.30	99.98	99.65	**99.87**	**97.9**
AIs	99.78	99.72	100	99.86	**99.95**	**98.0**

5 Experiment

5.1 Main Study

To benchmark the effectiveness of the watermark, we primarily report the discriminator's performance across 1000 generated images from all units of each protected dataset in the black-box validation

3

scenario of AI mimicry, utilizing metrics of Accuracy, Precision, Recall, and F1-Score (i.e., All indicate the Macro-average, Macro-average $= \frac{1}{C} \sum_{i=1}^{C}$ metric). As shown in Table 1, all of our evaluation metrics are above 99%, which demonstrates the superior performance of our model.
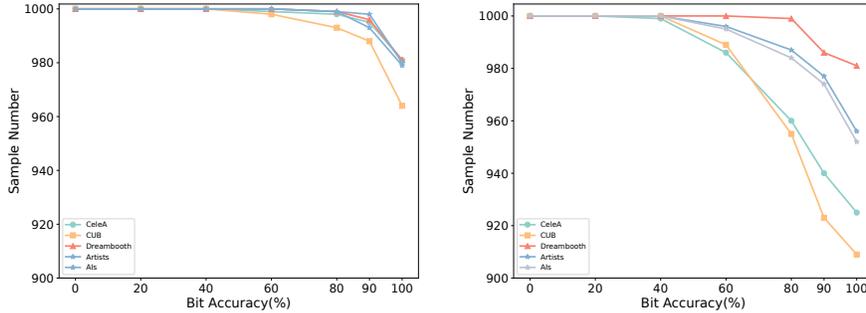
Figure 2: With the increase in bit accuracy, the trend of the number of validation samples that surpass the predicted bit accuracy threshold evolves. The left subplot illustrates the model's performance with Disentanglement, and the right subplot shows the performance without Disentanglement.

Meanwhile, We compared the impact of the Disentangled style domain on model performance across five benchmark datasets. In figure 1 and 2, we found that there has been a decline to varying degrees in the two metrics of $Avg\ acc$ and $t@k@100\%wd$.

Additionally, we employ the current SOTA (DIAGNOSIS[2]) for dataset protection through the backdoor-based method. The evaluation metrics utilized are True Positive (TP), True Negative (TN), and Attack Success Rate (ASR), as implemented by DIAGNOSIS. The results are shown in table 2.

Table 2: Main Result: Comparison of results between DIAGNOSIS and ours

Method	TP	TN	ASR (%)	Avg acc(%)	$t@k@100\%wd$ (%)
DIAGNOSIS	993	7	99.3	-	-
Ours	999	1	99.9	99.72	98

Post-tracking ownership: Post-tracking ownership refers to the process of claiming copyright ownership when the owner discovers suspicious models or images. Due to backdoors in mimic models that have been stolen and not timely injected, effective copyright claims cannot be made. The results are shown in table 3.

Table 3: Post-tracking ownership: Comparison of results between DIAGNOSIS and ours

Method	TP	TN	ASR (%)	Avg acc(%)	$t@k@100\%wd$ (%)
DIAGNOSIS	2	998	0.2	-	-
Ours	999	1	99.9	99.69	94.7

5.2 Robustness Study

We use the watermark removal method[3] to attack baseline watermarking schemes and ours. For attacks using variational autoencoders, we evaluate the pre-trained image compression models: Cheng2020 [4]. The compression factors are set to 3. For diffusion model attacks, we use stable diffusion 2.0. The number of noise steps is set to 60. we chose Avg acc (average watermark accuracy), Detect Acc (percentage of images where decoded bits exceed the detection threshold 0.65), and $t@k@100\%wd$ as the evaluation metrics for watermark robustness. The result is in 4. Our method achieves an average accuracy of 97.93% and 95.81%, with a detection accuracy of 100% and $t@k@100\%wd$ of 91.5% and 87.2% under VAE and Diffusion attacks, respectively. In contrast, other methods like DCT-DWT-SVD, RivaGan, and SSL show significantly lower performance. From the

4

results, our performance significantly surpasses other watermarking schemes after being subjected to watermark removal attacks.

Table 4: The Robustness study.

Method	Removal Attack Instance	Avg acc(%)	Detect Acc (%)	$k@t@100\%wd$ (%)
DCT-DWT-SVD	VAE attack	50.17	2.0	0.0
	Diffusion attack	54.41	2.8	0.0
RivaGan	VAE attack	60.71	6.2	0.0
	Diffusion attack	58.23	1.8	0.0
SSL	VAE attack	62.92	15.6	0.0
	Diffusion attack	63.21	16.3	0.0
Ours	VAE attack	**97.93**	**100**	**91.5**
	Diffusion attack	95.81	100	87.2

5.3 Ablation Study

We have incorporated a disentangled style domain into the ablation analysis to further investigate the robustness of our study. In table 5, We noticed a degradation in the robustness of our method against a range of adversarial attacks in the absence of the disentangled style domain.

Table 5: The ablation study specifically addresses the role of the Disentangled style domain in enhancing robustness.

128-bit	w\ Disentangle	The sample counts within each range of watermark distribution						$Avg\ acc$ (%) ↓	$t@k@100\%wd$ (%) ↓
		0-20%	20-40%	40-60%	60-80%	80-90%	90-100%		
z-watermarking	×	0	0	0	4	9	977	99.20	95.6
	✓	0	0	0	1	6	993	**99.87**	**97.9**
Second-stage Fine-tune	×	0	1	11	28	35	925	98.12	89.5
	✓	0	0	6	9	7	975	**99.13**	**93.3**
Mixed Clean Fine-tune	×	0	1	29	169	154	647	93.28	60.9
	✓	0	1	11	29	35	944	**99.04**	**92.2**
Latent Attack	×	0	1	44	139	154	662	91.23	62.9
	✓	0	0	13	19	24	925	**95.81**	**87.2**
Prompt Attack	×	0	15	118	186	179	613	90.01	52.7
	✓	0	0	95	9	36	860	**95.81**	**76.7**
Contrast	×	0	1	42	156	144	657	92.27	63.3
	✓	0	0	8	9	11	972	**99.01**	**92.2**
JPEG	×	0	2	41	192	161	604	78.53	60.2
	✓	0	0	8	10	14	968	**98.97**	**91.6**
GaussianBlur	×	0	0	70	162	145	623	90.67	60.7
	✓	0	0	11	17	15	957	**98.50**	**89.8**
Brightness	×	0	0	44	178	162	616	90.32	60.2
	✓	0	0	24	22	19	935	**97.63**	**88.1**
CenterCrop	×	0	13	320	330	173	164	78.53	16.4
	✓	0	0	43	82	68	805	**94.82**	**69.9**
Hue	×	0	64	351	395	153	123	77.31	11.2
	✓	0	0	37	80	50	833	**94.44**	**68.6**
Rotation	×	0	13	398	455	75	59	73.96	4.1
	✓	0	17	294	415	105	169	**83.66**	**14.8**

6 Analysis of Framework Efficiency

Our framework is divided into three stages: registration, computation, and inference. **In the registration stage**, data owners register their identifier z and corresponding watermark with a third-party regulatory body. **In the computation stage**, the third party uses our algorithm to perform computations and store the results after receiving the registration list. **In the inference stage**, the framework performs copyright verification on suspicious samples and models. We analyze the efficiency of the framework as follows: On one hand, in terms of resource consumption, during the computation stage, using a single 3090 GPU, the average computation time per user is 1 minute, with proxy sample computations averaging 3-5 minutes and memory usage approximately 3MB. In the inference stage, the average inference time for 1000 users ranges from 30 to 100 milliseconds (i.e., 0.065 milliseconds per user). On the other hand, regarding copyright tracing accuracy, the ASR metric is close to 100%, with an error rate controlled below 0.1%, and the average watermark accuracy exceeds 99%. Of 1000 suspicious AI mimic samples, about 97% can be successfully verified and traced through judicial

5

proceedings (as indicated by the $t@k@100wd\%$ metric mentioned in this paper). Overall, Users such as artists only need to register the identifier z and the corresponding watermark with the third party. The design and complexity of the algorithmic framework ensure the security, rigor, and accuracy of copyright protection. Overall, our framework demonstrates its practicality in judicial security validation, resource consumption, and efficiency.

References

[1] Shawn Shan, Jenna Cryan, Emily Wenger, Haitao Zheng, Rana Hanocka, and Ben Y Zhao. Glaze: Protecting artists from style mimicry by {Text-to-Image} models. In 32nd USENIX Security Symposium (USENIX Security 23), pages 2187–2204, 2023.

[2] Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N Metaxas, and Shiqing Ma. Diagnosis: Detecting unauthorized data usages in text-to-image diffusion models. In The Twelfth International Conference on Learning Representations, 2023.

[3] Xuandong Zhao, Kexun Zhang, Zihao Su, Saastha Vasan, Ilya Grishchenko, Christopher Kruegel, Giovanni Vigna, Yu-Xiang Wang, and Lei Li. Invisible image watermarks are provably removable using generative ai. arXiv preprint arXiv:2306.01953, 2023.

[4] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. Gan-leaks: A taxonomy of membership inference attacks against generative models. In Proceedings of the 2020 ACM SIGSAC conference on computer and communications security, pages 343–362, 2020.

6