Image Copy Detection for Diffusion Models

Wenhao Wang[1], **Yifan Sun**[2]*, **Zhentao Tan**[2], **Yi Yang**[3]
[1]University of Technology Sydney [2]Baidu Inc. [3]Zhejiang University

Figure 1: Some generated images (top) from diffusion models replicates the contents of existing images (bottom). The existing (matched) images are from LAION-Aesthetics [1]. The diffusion models include both commercial and open-source ones.

Abstract

Images produced by diffusion models are increasingly popular in digital artwork and visual marketing. However, such generated images might replicate content from existing ones and pose the challenge of content originality. Existing Image Copy Detection (ICD) models, though accurate in detecting hand-crafted replicas, overlook the challenge from diffusion models. This motivates us to introduce ICDiff, the first ICD specialized for diffusion models. To this end, we construct a Diffusion-Replication (D-Rep) dataset and correspondingly propose a novel deep embedding method. D-Rep uses a state-of-the-art diffusion model (Stable Diffusion V1.5) to generate $40,000$ image-replica pairs, which are manually annotated into 6 replication levels ranging from 0 (no replication) to 5 (total replication). Our method, PDF-Embedding, transforms the replication level of each image-replica pair into a probability density function (PDF) as the supervision signal. The intuition is that the probability of neighboring replication levels should be continuous and smooth. Experimental results show that PDF-Embedding surpasses protocol-driven methods and non-PDF choices on the D-Rep test set. Moreover, by utilizing PDF-Embedding, we find that the replication ratios of well-known diffusion models against an open-source gallery range from 10% to 20%. The project is publicly available at https://icdiff.github.io/.

1 Introduction

Diffusion models have gained popularity due to their ability to generate high-quality images. A phenomenon accompanying this trend is that these generated images might replicate content from

*Corresponding Author.

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

https://doi.org/10.52202/079017-0460

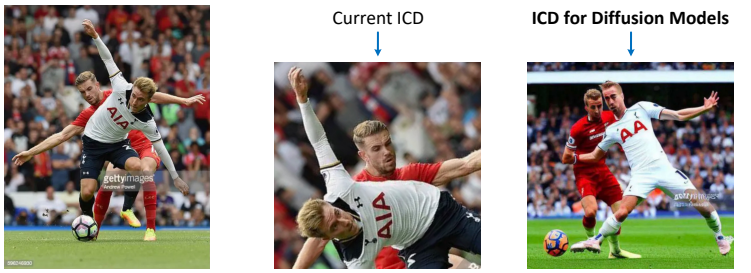

Figure 2: The comparison between current ICD with the ICDiff. The current ICD focuses on detecting edited copies generated by transformations like horizontal flips, random rotations, and random crops. In contrast, the ICDiff aims to detect replication generated by diffusion models, such as Stable Diffusion [2]. (Source of the original image: Lawsuit from Getty Images.)

existing ones. In Fig. 1, we choose six well-known diffusion models [3, 4, 5, 6, 7, 8] to illustrate this replication phenomenon. The content replication is acceptable for some (fair) use while interest holders may regard others as copyright infringement [9, 10, 11]. This paper leaves this dispute alone, and focuses a scientific problem: *How to identify the content replication brought by diffusion models?*

Image Copy Detection (ICD) provides a general solution to the above demand: it identifies whether an image is copied from a reference gallery after being tampered with. However, the current ICD methods are trained using hand-crafted image transformations (*e.g.*, horizontal flips, random rotations, and random crops) and overlook the challenge from diffusion models. Empirically, we find existing ICD methods can be easily confused by diffusion-generated replicas (as detailed in Table 3). We infer it is because the tamper patterns underlying diffusion-generated replicas (Fig. 2 right) are different from hand-crafted ones (Fig. 2 middle), yielding a considerable pattern gap.

In this paper, we introduce ICDiff, the first ICD specialized for diffusion-generated replicas. Our efforts mainly involve building a new ICD dataset and proposing a novel deep embedding method.

• **A Diffusion Replication (D-Rep) dataset.** D-Rep consists of $40,000$ image-replica pairs, in which each replica is generated by a diffusion model. Specifically, the images are from LAION-Aesthetic V2 [1], while their replicas are generated by Stable Diffusion V1.5 [12]. To make the replica generation more efficient, we search out the text prompts (from DiffusionDB [13])that are similar to the titles of LAION-Aesthetic V2 images, input these text prompts into Stable Diffusion V1.5, and generate many redundant candidate replicas. Given these candidate replicas, we employ human annotators to label the replication level of each generated image against a corresponding LAION-Aesthetic image. The annotation results in $40,000$ image-replica pairs with 6 replication levels ranging from 0 (no replication) to 5 (total replication). We divide D-Rep into a training set with 90% $(36,000)$ pairs and a test set with the remaining 10% $(4,000)$ pairs.

• **A novel method named PDF-Embedding.** The ICD methods rely on deep embedding learning at their core. In the deep embedding space, the replica should be close to its original image and far away from other images. Compared with popular deep embedding methods, our PDF-Embedding learns a Probability-Density-Function between two images, instead of a similarity score. More concretely, PDF-Embedding transforms the replication level of each image-replica pair into a PDF as the supervision signal. The intuition is that the probability of neighboring replication levels should be continuous and smooth. For instance, if an image-replica pair is annotated as level-3 replication, the probabilities for level-2 and level-4 replications should also not be significantly low.

PDF-Embedding predicts the probability scores on all the replication levels simultaneously in two steps: 1) extracting 6 feature vectors in parallel from both the real image and its replica, respectively and 2) calculating 6 inner products (between two images) to indicate the probability score at 6 corresponding replication levels. The largest-scored entry indicates the predicted replication level. Experimentally, we prove the effectiveness of our method by comparing it with popular deep embedding models and protocol-driven methods trained on our D-Rep. Moreover, we evaluate the replication of six famous diffusion models and provide a comprehensive analysis.

In conclusion, our key contributions are as follows:

1. We propose a timely and important ICD task, *i.e*, Image Copy Detection for Diffusion Models (ICDiff), designed specifically to identify the replication caused by diffusion models.
2. We build the first ICDiff dataset and introduce PDF-Embedding as a baseline method. PDF-Embedding transforms replication levels into probability density functions (PDFs) and learns a set of representative vectors for each image.
3. Extensive experimental results demonstrate the efficiency of our proposed method. Moreover, we discover that between 10% to 20% of images generated by six well-known diffusion models replicate contents of a large-scale image gallery.

2 Related Works

2.1 Existing Image Copy Detection Methods

Current ICD methods try to detect replications by learning the invariance of image transformations. For example, ASL [14] considers the relationship between image transformations and hard negative samples. AnyPattern [20] and PE-ICD [21] build benchmarks and propose solutions that focus on novel patterns in real-world scenarios. SSCD [15] reveals that self-supervised contrastive training inherently relies on image transformations, and thus adapts InfoNCE [16] by introducing a differential entropy regularization. BoT [17] incorporates approximately ten types of image transformations, combined with various tricks, to train an ICD model. By increasing the intensity of transformations gradually, CNNCL [18] successfully detects hard positives using a simple contrastive loss and memory bank. EfNet [19] ensembles models trained with different image transformations to boost the performance. In this paper, we discover that capturing the invariance of image transformations is ineffective for detecting copies generated by diffusion models. Consequently, we manually label a new dataset and train a specialized ICD model.

2.2 Replication in Diffusion Models

Past research has explored the replication problems associated with diffusion models. The study by [10] questions if diffusion models generate unique artworks or simply mirror the content from their training data. Research teams from Google, as highlighted in [22], note that diffusion models reveal their training data during content generation. Other studies prevent generating replications from the perspectives of both training diffusion models [23, 24, 25, 26, 27] and copyright holders [28, 29, 30]. Some experts, such as those in [24], find that the replication of data within training sets might be a significant factor leading to copying behaviors in diffusion models. To address this, [31] proposes an algorithmic chain to de-duplicate the training sources like LAION-2B [1]. In contrast to these efforts, our ICDiff offers a unique perspective. Specifically, unlike those that directly using existing image descriptors (such as those from CLIP [32] and SSCD [15]), we manually-label a dataset and develop a specialized ICD algorithm. By implementing our method, the analytical tools and preventative strategies proposed in existing studies may achieve greater efficacy.

3 Benchmark

This section introduces the proposed ICD for diffusion models (ICDiff), including our dataset (D-Rep) and the corresponding evaluation protocols.

3.1 D-Rep Dataset

Current ICD [33, 34, 35, 14, 21, 20] primarily focuses on the replica challenges brought by hand-crafted transformations. In contrast, our ICDiff aims to address the replication issues caused by diffusion models [3, 4, 5, 6, 7, 8]. To facilitate ICDiff research, we construct D-Rep dataset, which is characterized for diffusion-based replication (See Fig. 3 and the Appendix (Section A) for the examples of diffusion-based replica). The construction process involves generating candidate pairs followed by manual labeling.

Generating candidate pairs. It consists of (1) selecting the top $40,000$ most similar prompts and titles: this selection provides an abundant image-replica pair source. In detail, we use the Sentence Transformer [36] to encode the 1.8 million real-user generated prompts from DiffusionDB [13] and

https://doi.org/10.52202/079017-0460

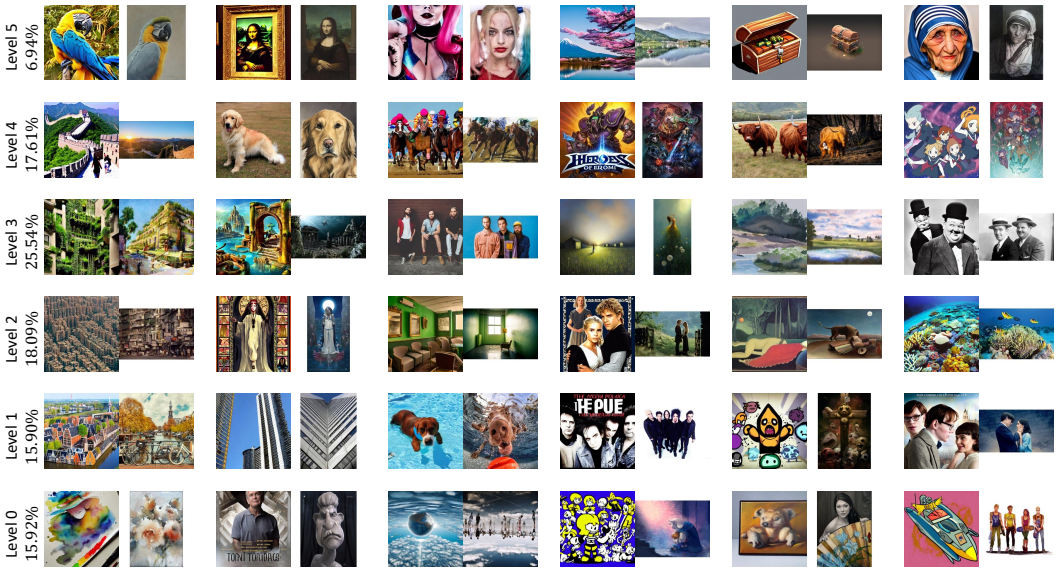

Figure 3: The demonstration of the manual-labeled D-Rep dataset. The percentages on the left show the proportion of images with a particular level.

the 12 million image titles from LAION-Aesthetics V2 6+ [1], and then utilize the computed cosine similarities to compare; (2) obtaining the candidate pairs: the generated images are produced using the prompts with Stable Diffusion V1.5 [12], and the real images are fetched based on the titles.

Manual labeling. We generally follow the definition of replication in [10] and further define six levels of replication (0 to 5). A higher level indicates a greater degree that the generated image replicates the real image. Due to the complex nature of diffusion-generated images, we use multiple levels instead of the binary levels used in [10], which employed manual-synthetic datasets as shown in their Fig. 2. We then train ten professional labelers to assign these levels to the $40,000$ candidate pairs: Initially, we assign $4,000$ image pairs to each labeler. If labelers are confident in their judgment of an image pair, they will directly assign a label. Otherwise, they will place the image pair in an undecided pool. On average, each labeler has about 600 undecided pairs. Finally, for each undecided pair, we vote to reach a final decision. For example, if the votes for an undecided pair are 2, 2, 2, 3, 3, 3, 3, 3, 4, 4, the final label assigned is 3. Given the complexity of this labeling task, it took both the labelers and our team one month to finish the process. To maintain authenticity, we did not pre-determine the proportion of each score. The resulting proportions are on the left side of Fig. 3.

3.2 Evaluation Protocols

To evaluate ICD models on the D-Rep dataset, we divide the dataset into a 90/10 training/test split and design two evaluation protocols: Pearson Correlation Coefficient (PCC) and Relative Deviation (RD).

Pearson Correlation Coefficient (PCC). The PCC is a measure used to quantify the linear relationship between two sequences. When PCC is near 1 or -1, it indicates a strong positive or negative relationship. If PCC is near 0, there's little to no correlation between the sequences. Herein, we consider two sequences, the predicted replication level $\boldsymbol{s}^p = (s_1^p, s_2^p, \ldots, s_n^p)$ and the ground-truth $\boldsymbol{s}^l = \left(s_1^l, s_2^l, \ldots, s_n^l\right)$ (n is the number of test pairs). The PCC for ICDiff is defined as:

$$\text{PCC} = \frac{\sum_{i=1}^{n} \left(s_i^p - \bar{\boldsymbol{s}^p}\right)\left(s_i^l - \bar{\boldsymbol{s}^l}\right)}{\sqrt{\sum_{i=1}^{n} \left(s_i^p - \bar{\boldsymbol{s}^p}\right)^2} \times \sqrt{\sum_{i=1}^{n} \left(s_i^l - \bar{\boldsymbol{s}^l}\right)^2}}, \tag{1}$$

where $\bar{\boldsymbol{s}^p}$ and $\bar{\boldsymbol{s}^l}$ are the mean values of $\boldsymbol{s}^p$ and $\boldsymbol{s}^l$, respectively.

https://doi.org/10.52202/079017-0460

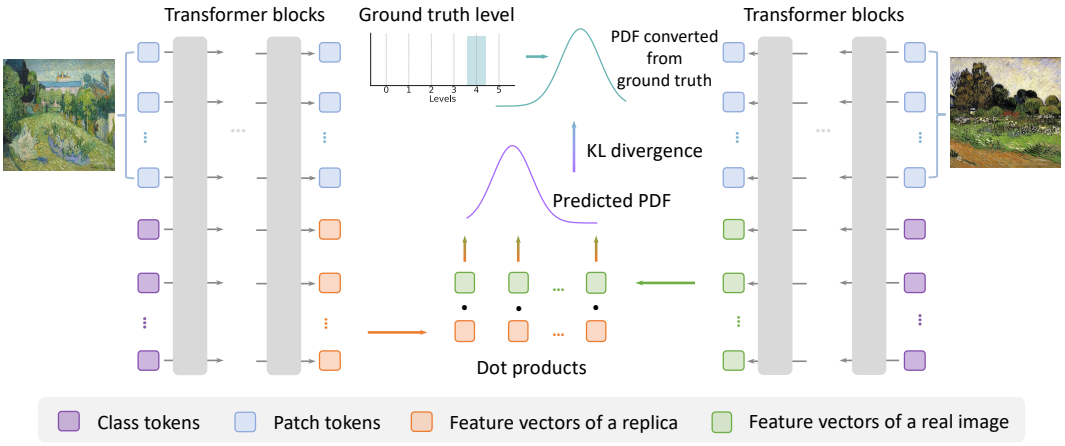

Figure 4: The demonstration of the proposed PDF-Embedding. Initially, PDF-Embedding converts manually-labeled replication levels into probability density functions (PDFs). To learn from these PDFs, we use a set of vectors as the representation of an image.

A limitation of the PCC is its insensitivity to global shifts. If all the predictions differ from their corresponding ground truth with the same shift, the PCC does not reflect such a shift and remains large. To overcome this limitation, we propose a new metric called the Relative Deviation (RD).

Relative Deviation (RD). We use RD to quantify the normalized deviation between the predicted and the labeled levels. By normalizing against the maximum possible deviation, RD provides a measure of how close the predictions are to the labeled levels on a scale of 0 to 1. The RD is calculated by

$$\mathrm{RD} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\left|s_i^p - s_i^l\right|}{\max\left(N - s_i^l, s_i^l\right)}\right), \tag{2}$$

where N is the highest replication level in our D-Rep.

The Preference for RD over Absolute One. Here we show the preference for employing RD over the absolute one through two illustrative examples. We denote the relative and absolute deviation of the ith test pair as: $S_i = \frac{\left|s_i^p - s_i^l\right|}{\max\left(N - s_i^l, s_i^l\right)}$, and $T_i = \frac{\left|s_i^p - s_i^l\right|}{N}$.

(1) For a sample with $s_i^l = 3$, if $s_i^p = 3$, both S_i and T_i equal 0; however, if $s_i^p = 0$ (representing the worst prediction), $S_i = 1$ and $T_i = 0.6$. Here, S_i adjusts the worst prediction to a value of 1.

(2) In the first scenario, where $s_i^l = 3$ and $s_i^p = 0, S_i = 1$ and $T_i = 0.6$. In the second scenario, where $s_i^l = 5$ and $s_i^p = 2, S_i = 0.6$ and $T_i = 0.6$. For both cases, T_i remains the same at 0.6, whereas S_i values differ. Nevertheless, the two scenarios are distinct: in the first, the prediction cannot deteriorate further; in the second, it can. The value of S_i accurately captures this distinction, whereas T_i does not.

4 Method

This section introduces our proposed PDF-Embedding for ICDiff. PDF-Embedding converts each replication level into a probability density function (PDF). To facilitate learning from these PDFs, we expand the original representative vector into a set of vectors. The demonstration of PDF-Embedding is displayed in Fig. 4.

4.1 Converting Level to Probability Density

Given an replication level $s^l \in \boldsymbol{s}^l$, we first normalize it into $p^l = s^l / max(\boldsymbol{s}^l)$. Then we transfer p^l into a PDF denoted as $g(x)$ [2], where $x \in [0, 1]$ indicates each possible normalized level. The

[2] Although we acknowledge that the random variables in this context are discrete, we still utilize the term "PDF" to effectively communicate our intuition and present our method under ideal conditions.

https://doi.org/10.52202/079017-0460

intuition is that the probability distribution of neighboring replication levels should be continuous and smooth. The function $g(x)$ must satisfy the following conditions: (1) $\int_0^1 g(x)dx = 1$, ensuring that $g(x)$ is a valid PDF; (2) $g(x) \geq 0$, indicating the non-negativity of the PDF; and (3) $g(x) \leq g(p^l)$, which implies that the density is maximized at the normalized level p^l. We use three different implementations for $g(x)$:

Gaussian:

$$g(x \mid A, \mu, \sigma) = A \cdot \exp\left(-\frac{(x-\mu)^2}{2 \cdot \sigma^2}\right), \tag{3}$$

linear:

$$g(x \mid A, \mu, \beta) = A - \beta \cdot |x - \mu|, \tag{4}$$

and exponential:

$$g(x \mid A, \mu, \lambda) = A \cdot \lambda \cdot \exp(-\lambda \cdot |x - \mu|), \tag{5}$$

where: A is the amplitude, and μ is the center; σ is the standard deviation of Gaussian function, β is the slope of linear function, and λ is the spread of exponential function.

The performance achieved with various converted PDFs is illustrated in the experimental section. For additional details, please see the Appendix (Section B), which includes (1) **the methodology for calculating distribution values**, (2) **the visualization of the learned distributions corresponding to different image pairs**, and (3) **an analysis of the deviation rate from peak values**.

4.2 Representing an Image as a Set of Vectors

To facilitate learning from the converted PDFs, we utilize a Vision Transformer (ViT) [37] to represent an image as a set of vectors. Let's denote the patch tokens from a real image as $\mathbf{X}_r^0$ and from a generated image as $\mathbf{X}_g^0$, the ViT model as f, and the number of layers in ViT as L. The feed-forward process can be expressed as:

$$\begin{aligned} \left[\mathbf{C}_r^L, \mathbf{X}_r^L\right] &= f\left(\left[\mathbf{C}^0, \mathbf{X}_r^0\right]\right), \\ \left[\mathbf{C}_g^L, \mathbf{X}_g^L\right] &= f\left(\left[\mathbf{C}^0, \mathbf{X}_g^0\right]\right), \end{aligned} \tag{6}$$

where $\mathbf{C}^0$ is a set of class tokens; $\mathbf{C}_r^L$ is a set of representative vectors for the real images, consisting of vectors $c_{0,r}^L, c_{1,r}^L, \ldots, c_{N,r}^L$, and $\mathbf{C}_g^L$ is a set of representative vectors for the generated images, consisting of vectors $c_{0,g}^L, c_{1,g}^L, \ldots, c_{N,g}^L$; N is the highest replication level.

Therefore, we can use another PDF $h\,(x)$ ($x \in [0, 1]$) to describe the predicted replication between two images by

$$h(x) = \mathbf{C}_r^L \cdot \mathbf{C}_g^L, \tag{7}$$

which expands to:

$$h(x) = \left[\left\langle \mathbf{c}_{0,r}^L, \mathbf{c}_{0,g}^L \right\rangle, \left\langle \mathbf{c}_{1,r}^L, \mathbf{c}_{1,g}^L \right\rangle, \ldots, \left\langle \mathbf{c}_{N,r}^L, \mathbf{c}_{N,g}^L \right\rangle\right], \tag{8}$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity.

For training, recalling the PDF $g(x)$ derived from the level, we define the final loss using the Kullback-Leibler (KL) divergence:

$$\mathcal{L} = D_{KL}(g \| h) = \int_0^1 g(x) \log\left(\frac{g(x)}{h(x)}\right) dx, \tag{9}$$

which serves as a measure of the disparity between two probability distributions. Additionally, in the Appendix (Section C), we demonstrate what the network captures during its learning phase.

During testing, the normalized level between two images is denoted by $\hat{p}^l$, satisfying $h(x) \leq h(\hat{p}^l)$. As illustrated in Eqn. 8, $h(x)$ in practice is discrete within the interval $[0, 1]$. Consequently, the resulting level is

$$j = \operatorname{argmax} h(x), \tag{10}$$

and the normalized level is quantified as $\frac{j}{N}$.

Table 1: The performance of publicly available models and our PDF-Embedding on the D-Rep. For qualitative results, please refer to Section E in the Appendix.

Class	Method	PCC (%) ↑	RD (%) ↓
Vision-language Models	SLIP [41]	31.8	49.7
	BLIP [42]	34.8	41.6
	ZeroVL [43]	36.3	36.5
	CLIP [32]	36.8	35.8
	GPT-4V [44]	47.3	38.7
Self-supervised Learning Models	SimCLR [45]	7.2	49.4
	MAE [46]	20.7	67.6
	SimSiam [47]	33.5	45.4
	MoCov3 [48]	35.7	40.3
	DINOv2 [49]	39.0	32.9
Supervised Pre-trained Models	EfficientNet [50]	24.0	59.3
	Swin-B [51]	32.5	38.4
	ConvNeXt [52]	33.8	36.0
	DeiT-B [40]	35.3	41.7
	ResNet-50 [53]	37.5	34.5
Current ICD Models	ASL [14]	5.6	78.1
	CNNCL [18]	19.1	51.7
	SSCD [15]	29.1	62.3
	EfNet [19]	30.5	62.8
	BoT [17]	35.6	53.8

5 Experiments

5.1 Training Details

We implement our PDF-Embedding using PyTorch [38] and distribute its training over 8 A100 GPUs. The ViT-B/16 [37] serves as the backbone and is pre-trained on the ImageNet dataset [39] using DeiT [40], unless specified otherwise. We resize images to a resolution of 224×224 before training. A batch size of 512 is used, and the total training epochs is 25 with a cosine-decreasing learning rate.

5.2 Challenge from the ICDiff Task

This section benchmarks popular public models on our D-Rep test dataset. As Table 3 shows, we conduct experiments extensively on vision-language models, self-supervised models, supervised pre-trained models, and current ICD models. We employ these models as feature extractors and calculate the cosine similarity between pairs of image features (except for GPT-4V Turbo [44], see Section D in the Appendix for the implementation of it). For the computation of PCC and RD, we adjust the granularity by scaling the computed cosine similarities by a factor of N. In the Appendix (Section E), we further present the concrete similarities predicted by these models and provide corresponding analysis. We observe that: (1) the large multimodal model GPT-4V Turbo [44] performs best in PCC, while the self-supervised model DINOv2 [49] excels in RD. This can be attributed to their pre-training on a large, curated, and diverse dataset. Nevertheless, their performance remains somewhat limited, achieving only 47.3% in PCC and 32.9% in RD. This underscores that even the best publicly available models have yet to effectively address the ICDiff task. (2) Current ICD models, like SSCD [15], which are referenced in analysis papers [10, 24] discussing the replication issues of diffusion models, indeed show poor performance. For instance, SSCD [15] registers only 29.1% in PCC and 62.3% in RD. Even the more advanced model, BoT [17], only manages 35.6% in PCC and 53.8% in RD. These results underscore the need for a specialized ICD method for diffusion models. Adopting our specialized ICD approach will make their subsequent analysis more accurate and convincing. (3) Beyond these models, we also observe that others underperform on the ICDiff task. This further emphasizes the necessity of training a specialized ICDiff model.

5.3 The Effectiveness of PDF-Embedding

This section demonstrates the effectiveness of our proposed PDF-Embedding by (1) contrasting it against protocol-driven methods and non-PDF choices on the D-Rep dataset, (2) comparing between different distributions, and (3) comparing with other models in generalization settings.

https://doi.org/10.52202/079017-0460

Table 2: Our method demonstrates performance superiority over others.

Method	PCC (%) ↑	RD (%) ↓	Train ($s/iter$) ↓	Infer (s/img) ↓	Match ($s/pair$) ↓
Enlarging PCC	54.4	40.1	0.293	2.02×10^{-3}	1.02×10^{-9}
Reducing RD	15.1	29.9	0.294		
Regression	40.3	28.1	0.292		
One-hot Label	37.6	43.3	0.310	2.07×10^{-3}	6.97×10^{-9}
Label Smoothing	35.0	36.1			
Ours (Gaussian)	53.7	**24.0**	0.310	2.07×10^{-3}	6.97×10^{-9}
Ours (Linear)	54.0	24.6			
Ours (Exp.)	**56.3**	25.6			

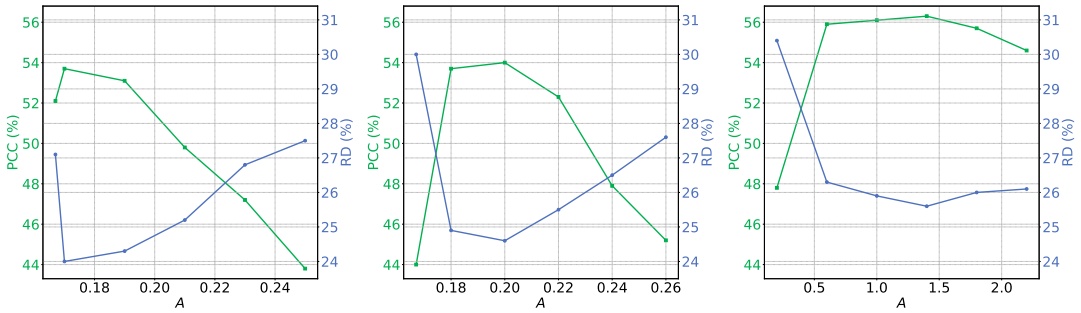

Figure 5: The comparison of different PDFs: Gaussian (left), linear (middle), and exponential (right). "A" is the amplitude in each PDF function (Eqn. 3 to Eqn. 5).

Comparison against protocol-driven methods. Since we employ PCC and RD as the evaluation protocols, a natural embedding learning would be directly using these protocols as the optimization objective, *i.e.*, enlarging PCC and reducing RD. Moreover, we add another variant of "reducing RD", *i.e.*, reducing the absolute deviation $\left|s_i^p - s_i^l\right|$ in a regression manner. The comparisons are summarized in Table 2, from which we draw three observations as below: (1) Training on D-Rep with the protocol-driven method achieves good results on their specified protocol but performs bad for the other. While "Enlarging PCC" attains a commendable PCC, its RD of 40.1% indicates large deviation from the ground truth. "Reducing RD" or "Reducing Deviation" shows a relatively good RD (28.1%); however, they exhibit small PCC values that indicate low linear consistency. (2) Our proposed PDF-Embedding surpasses these protocol-driven methods in both PCC and RD. Compared against "Enlarging PCC", our method improves PCC by 1.6% and decreases RD by 16.1%. Besides, our method achieves $+16.0\%$ PCC and -4.1% RD compared against "Reducing RD" and "Reducing Deviation". (3) The computational overhead introduced by our method is negligible. First, compared to other options, our method only increases the training time by 5.8%. Second, our method introduces minimal additional inference time. Third, while our method requires a longer matching time, its magnitude is close to 10^{-9}, which is negligible when compared to the inference time's magnitude of 10^{-3}. Further discussions on the matching time in real-world scenarios can be found in Section 5.4.

Comparison against two non-PDF methods. In Table 2, we also show the experimental results of our method under two standard supervising signals, *i.e.*, "One-hot Label" and "Label Smoothing ($\epsilon = 0.5$)". In comparison, our PDF-Embedding using PDFs gains significant superiority, *e.g.*, using exponential PDF is better than label smoothing by $+21.3\%$ PCC and -10.5% RD. This superiority validates our intuition that neighboring replication levels should be continuous and smooth.

Comparison between different PDF implementations. We compare between three different PDF implementations for the proposed PDF-Embedding in Fig. 5. We observe that: (1) The exponential (convex) function benefits the PCC metric, whereas the Gaussian (concave) function favors the RD metric. The performance of the linear function, which lacks curvature, falls between that of the convex and concave functions. (2) Our method demonstrates robust performance across various distributions, reducing the challenge of selecting an optimal parameter. For example, when using the exponential function, the performance remains high when A ranges from 0.6 to 1.8. (3) A model

https://doi.org/10.52202/079017-0460

Table 3: The experiments for "Generalizability to other datasets or diffusion models". The gray color indicates training and testing on the images generated by the same diffusion model.

Class	Method	SD1.5	Midjourney	DALL·E 2	DeepFloyd IF	New Bing	SDXL	GLIDE
Vision-language Models	SLIP [41]	0.685	0.680	0.668	0.710	0.688	0.718	0.699
	BLIP [42]	0.703	0.674	0.673	0.696	0.696	0.717	0.689
	ZeroVL [43]	0.578	0.581	0.585	0.681	0.589	0.677	0.707
	CLIP [32]	0.646	0.665	0.694	0.728	0.695	0.735	0.727
	GPT-4V [44]	0.661	0.655	0.705	0.731	0.732	0.747	0.744
Self-supervised Learning Models	SimCLR [45]	0.633	0.640	0.644	0.656	0.649	0.651	0.655
	MAE [46]	0.489	0.488	0.487	0.492	0.487	0.489	0.490
	SimSiam [47]	0.572	0.611	0.619	0.684	0.620	0.645	0.683
	MoCov3 [48]	0.585	0.526	0.535	0.579	0.541	0.554	0.599
	DINOv2 [49]	0.766	0.529	0.593	0.723	0.652	0.734	0.751
Supervised Pre-trained Models	EfficientNet [50]	0.116	0.185	0.215	0.241	0.171	0.210	0.268
	Swin-B [51]	0.334	0.387	0.391	0.514	0.409	0.430	0.561
	ConvNeXt [52]	0.380	0.429	0.432	0.543	0.433	0.488	0.580
	DeiT-B [40]	0.386	0.478	0.496	0.603	0.528	0.525	0.694
	ResNet-50 [53]	0.362	0.436	0.465	0.564	0.450	0.522	0.540
Current ICD Models	ASL [14]	0.183	0.231	0.093	0.122	0.048	0.049	0.436
	CNNCL [18]	0.201	0.311	0.270	0.347	0.279	0.358	0.349
	SSCD [15]	0.116	0.181	0.180	0.303	0.166	0.239	0.266
	EfNet [19]	0.133	0.265	0.267	0.438	0.249	0.340	0.349
	BoT [17]	0.216	0.345	0.346	0.477	0.338	0.401	0.489
Models Trained on D-Rep	Enlarging PCC	0.598	0.510	0.523	0.595	0.506	0.554	0.592
	Reducing RD	0.795	0.736	0.736	0.768	0.729	0.768	0.785
	Regression	0.750	0.694	0.705	0.739	0.704	0.721	0.744
	One-hot Label	0.630	0.376	0.400	0.562	0.500	0.548	0.210
	Label Smoothing	0.712	0.568	0.636	0.628	0.680	0.676	0.548
Ours	Gaussian PDF	0.787	0.754	0.784	0.774	0.774	0.780	0.776
	Linear PDF	0.822	0.758	0.798	0.794	0.782	0.794	0.790
	Exponential PDF	**0.831**	**0.814**	**0.826**	**0.804**	**0.802**	**0.818**	**0.794**

supervised by a smooth PDF outperforms that supervised by a steeper one (also see the corresponding distributions in Fig. 15 of the Appendix). That consists with our intuition again.

Our model has good generalizability compared to all other methods. Because the collection process of the images from some diffusion models (see Appendix F) differs from the process used to build the test set of our D-Rep dataset, it is difficult to label 6 levels for them and the proposed PCC and RD are not suitable. In the Table 3, we consider a quantitative evaluation protocol that measures the average similarity predicted by a model for given N image pairs, which are manually labeled with the highest level. When normalized to a range of 0 to 1, a larger value implies better predictions. This setting is practical because, in the real world, most people's concerns focus on where replication indeed occurs. We manually confirm 100 such pairs for each diffusion model. We draw three conclusions: (1) Our PDF-Embedding is more generalizable compared to all zero-shot solutions, such as CLIP, GPT4-V, and DINOv2; (2) Our PDF-Embedding still surpasses all other plausible methods trained on the D-Rep dataset in the generalization setting; (3) Compared against testing on SD1.5 (same domain), for the proposed PDF-Embedding, there is no significant performance drop on the generalization setting.

5.4 Simulated Evaluation of Diffusion Models

In this section, we simulate a scenario using our trained PDF-Embedding to evaluate popular diffusion models. We select 6 famous diffusion models, of which three are commercial, and another three are open source (See Section F in the Appendix for more details). We use the LAION-Aesthetics V2 6+ dataset [1] as the gallery and investigate whether popular diffusion models replicate it. When assessing the replication ratio of diffusion models, we consider image pairs rated at Level 4 and Level 5 to be replications.

https://doi.org/10.52202/079017-0460

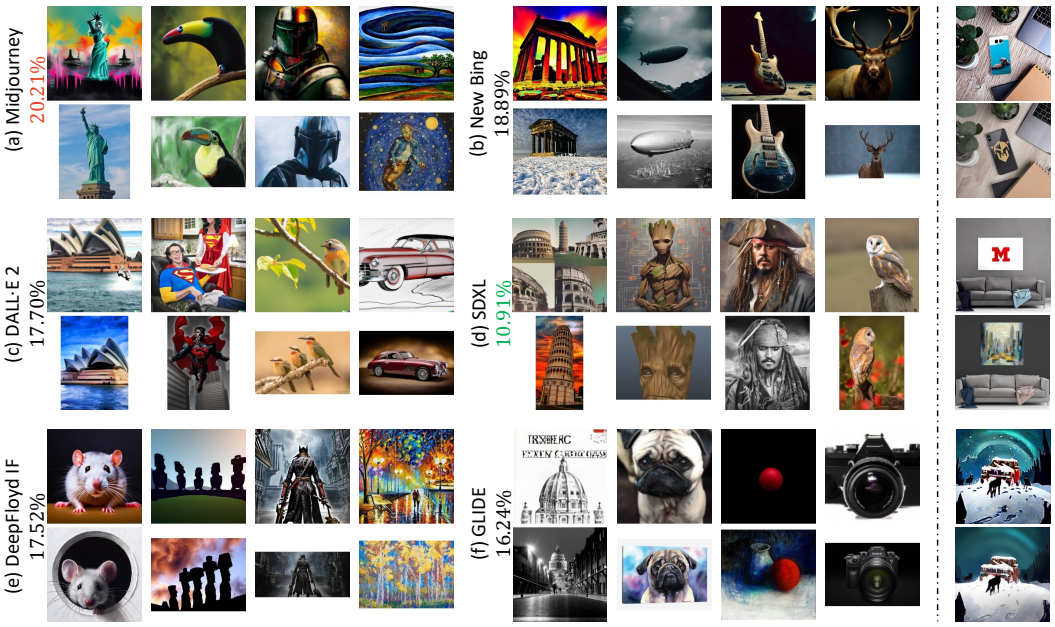

Figure 6: Left: Examples of diffusion-based replication fetched by our PDF-Embedding. The accompanying percentages indicate the replication ratio of each model. Right: Examples filtered by SSCD [15] in [10]. Compared to them, our results are more diverse: For example, the "Groot" generated by SDXL includes the whole body, whereas the original one features only the face; and the "Moai statues" created by DeepFloyd IF are positioned differently compared to the original image.

Evaluation results. Visualizations of matched examples and the replication ratios are shown in Fig. 6 (Left). For more visualizations, please refer to the Appendix (Section G). We observe that the replication ratios of these diffusion models roughly range between 10% and 20%. The most "aggressive" model is Midjourney [3] with a rate of 20.21%, whereas the "conservative" model is SDXL [6] at 10.91%. We also include an analysis of failure cases in the Appendix (Section H).

Efficiency analysis. Efficiency is crucial in real-world scenarios. A replication check might slow down the image generation speed of diffusion models. Our PDF-Embedding requires only 2.07×10^{-3} seconds for inference and an additional 8.36×10^{-2} seconds for matching when comparing a generated image against a reference dataset of 12 million images using a standard A100 GPU. This time overhead is negligible compared to the time required for generating (several seconds).

Intuitive comparison with another ICD model. In [10], SSCD [15] is used as a feature extractor to identify replication, as illustrated in Fig. 6 (Right). In comparison, our PDF-Embedding detects a higher number of challenging cases ("hard positives"). Despite visual discrepancies between the generated and original images, replication has indeed occurred.

6 Conclusion

This paper investigates a particular and critical Image Copy Detection (ICD) problem: Image Copy Detection for Diffusion Models (ICDiff). We introduce the first ICDiff dataset and propose a strong baseline called "PDF-Embedding". A distinctive feature of the D-Rep is its use of replication levels. The dataset annotates each replica into 6 different replication levels. The proposed PDF-Embedding first transforms the annotated level into a probability density function (PDF) to smooth the probability. To learn from the PDFs, our PDF-Embedding adopts a set of representative vectors instead of a traditional representative vector. We hope this work serves as a valuable resource for research on replication in diffusion models and encourages further research efforts in this area.

Disclaimer. The model described herein may yield false positive or negative predictions. Consequently, the contents of this paper should not be construed as legal advice.

https://doi.org/10.52202/079017-0460

References

[1] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[2] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021.

[3] Midjourney. Midjourney.com, 2022. Accessed: 2023-10-10.

[4] The new bing, 2023. Accessed: October 10, 2023.

[5] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022.

[6] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[7] Deep-floyd. If, 2023. Accessed: October 10, 2023.

[8] Alexander Quinn Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob Mcgrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, pages 16784–16804. PMLR, 2022.

[9] Katherine Lee, A Feder Cooper, and James Grimmelmann. Talkin"bout ai generation: Copyright and the generative-ai supply chain (the short version). In *Proceedings of the Symposium on Computer Science and Law*, pages 48–63, 2024.

[10] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Diffusion art or digital forgery? investigating data replication in diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6048–6058, 2023.

[11] Yuxin Wen, Yuchen Liu, Chen Chen, and Lingjuan Lyu. Detecting, explaining, and mitigating memorization in diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

[12] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[13] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hoover, and Duen Horng Chau. DiffusionDB: A large-scale prompt gallery dataset for text-to-image generative models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 893–911, Toronto, Canada, July 2023. Association for Computational Linguistics.

[14] Wenhao Wang, Yifan Sun, and Yi Yang. A benchmark and asymmetrical-similarity learning for practical image copy detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2672–2679, 2023.

[15] Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14532–14542, 2022.

[16] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[17] Wenhao Wang, Weipu Zhang, Yifan Sun, and Yi Yang. Bag of tricks and a strong baseline for image copy detection. *arXiv preprint arXiv:2111.08004*, 2021.

[18] Shuhei Yokoo. Contrastive learning with large memory bank and negative embedding subtraction for accurate copy detection. *arXiv preprint arXiv:2112.04323*, 2021.

[19] Sergio Manuel Papadakis and Sanjay Addicam. Producing augmentation-invariant embeddings from real-life imagery. *arXiv preprint arXiv:2112.03415*, 2021.

[20] Wenhao Wang, Yifan Sun, Zhentao Tan, and Yi Yang. Anypattern: Towards in-context image copy detection. In *arXiv preprint arXiv:2404.13788*, 2024.

[21] Wenhao Wang, Yifan Sun, and Yi Yang. Pattern-expandable image copy detection. In *International Journal of Computer Vision*, 2024.

[22] Nicolas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwag, Florian Tramer, Borja Balle, Daphne Ippolito, and Eric Wallace. Extracting training data from diffusion models. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 5253–5270, 2023.

 https://doi.org/10.52202/079017-0460

[23] Eric Zhang, Kai Wang, Xingqian Xu, Zhangyang Wang, and Humphrey Shi. Forget-me-not: Learning to forget in text-to-image diffusion models. *arXiv preprint arXiv:2303.17591*, 2023.

[24] Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. *Advances in Neural Information Processing Systems*, 2023.

[25] Nikhil Vyas, Sham Kakade, and Boaz Barak. Provable copyright protection for generative models. 2023.

[26] Anonymous. Copyright plug-in market for the text-to-image copyright protection. In *Submitted to The Twelfth International Conference on Learning Representations*, 2023. under review.

[27] Yunqing Zhao, Tianyu Pang, Chao Du, Xiao Yang, Ngai-Man Cheung, and Min Lin. A recipe for watermarking diffusion models. *arXiv preprint arXiv:2303.10137*, 2023.

[28] Yingqian Cui, Jie Ren, Yuping Lin, Han Xu, Pengfei He, Yue Xing, Wenqi Fan, Hui Liu, and Jiliang Tang. Ft-shield: A watermark against unauthorized fine-tuning in text-to-image diffusion models. *arXiv preprint arXiv:2310.02401*, 2023.

[29] Anthony Rhodes, Ram Bhagat, Umur Aybars Ciftci, and Ilke Demir. My art my choice: Adversarial protection against unruly ai. *arXiv preprint arXiv:2309.03198*, 2023.

[30] Yingqian Cui, Jie Ren, Han Xu, Pengfei He, Hui Liu, Lichao Sun, and Jiliang Tang. Diffusionshield: A watermark for copyright protection against generative diffusion models. *arXiv preprint arXiv:2306.04642*, 2023.

[31] Ryan Webster, Julien Rabin, Loic Simon, and Frederic Jurie. On the de-duplication of laion-2b. *arXiv preprint arXiv:2303.12733*, 2023.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[33] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM international conference on image and video retrieval*, pages 1–8, 2009.

[34] Matthijs Douze, Giorgos Tolias, Ed Pizzi, Zoë Papakipos, Lowik Chanussot, Filip Radenovic, Tomas Jenicek, Maxim Maximov, Laura Leal-Taixé, Ismail Elezi, et al. The 2021 image similarity dataset and challenge. *arXiv preprint arXiv:2106.09672*, 2021.

[35] Zoë Papakipos, Giorgos Tolias, Tomas Jenicek, Ed Pizzi, Shuhei Yokoo, Wenhao Wang, Yifan Sun, Weipu Zhang, Yi Yang, Sanjay Addicam, et al. Results and findings of the 2021 image similarity challenge. In *NeurIPS 2021 Competitions and Demonstrations Track*, pages 1–12. PMLR, 2022.

[36] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, 2019.

[37] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[40] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.

[41] Norman Mu, Alexander Kirillov, David Wagner, and Saining Xie. Slip: Self-supervision meets language-image pre-training. In *European Conference on Computer Vision*, pages 529–544. Springer, 2022.

[42] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, pages 12888–12900. PMLR, 2022.

[43] Quan Cui, Boyan Zhou, Yu Guo, Weidong Yin, Hao Wu, Osamu Yoshie, and Yubo Chen. Contrastive vision-language pre-training with limited resources. In *European Conference on Computer Vision*, pages 236–253. Springer, 2022.

[44] OpenAI. Gpt-4 technical report, 2023.

[45] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[46] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[47] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15750–15758, 2021.

[48] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[49] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[50] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[51] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[52] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[53] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

 https://doi.org/10.52202/079017-0460

A More examples of the D-Rep Dataset

We show more example image pairs for each level in Fig. 9 to Fig. 14.

B The Instantiations of PDFs

This section presents examples of PDFs derived from replication levels, focusing on three primary functions: Gaussian, linear, and exponential for our calculations, visualization, and analysis. Within the area close to the normalized level p^l, denoted as δ, the Gaussian function curves downwards making it concave, the linear function is straight with no curvature, and the exponential function curves upwards making it convex. These characteristics indicate the rate at which they deviate from their peak value: the Gaussian function changes slowly in the δ area, the linear function changes at a steady rate, and the exponential function changes rapidly in the δ area. A fast rate of change suggests the network learns from a sharp distribution, while a slow rate implies learning from a smooth distribution.

Gaussian function. Its general formulation is

$$g(x \mid A, \mu, \sigma) = A \cdot \exp\left(-\frac{(x-\mu)^2}{2 \cdot \sigma^2}\right), \tag{11}$$

where $A > 0$ is the amplitude (the height of the peak), $\mu \in [0, 1]$ is the mean or the center, and $\sigma > 0$ is the standard deviation. To satisfy the requirements of a PDF in Section 4.1, the following must hold:

$$\begin{gathered} \int_0^1 \left(A \cdot \exp\left(-\frac{(x-\mu)^2}{2 \cdot \sigma^2}\right)\right) dx = 1, \\ A \cdot \exp\left(-\frac{(x-\mu)^2}{2 \cdot \sigma^2}\right) \geq 0, \\ A \cdot \exp\left(-\frac{(x-\mu)^2}{2 \cdot \sigma^2}\right) \leq A \cdot \exp\left(-\frac{(p^l-\mu)^2}{2 \cdot \sigma^2}\right). \end{gathered} \tag{12}$$

From Eqn. 12, we have:

$$\mu = p^l. \tag{13}$$

In practice, with $x \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ being discrete, the equations become:

$$\begin{gathered} \sum_{x \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}} \left(A \cdot \exp\left(-\frac{(x-p^l)^2}{2 \cdot \sigma^2}\right)\right) = 1, \\ A \cdot \exp\left(-\frac{(x-p^l)^2}{2 \cdot \sigma^2}\right) \geq 0. \end{gathered} \tag{14}$$

Given a specific normalized level p^l and varying A, $g(x \mid A, \mu, \sigma)$ values are computed for different x using numerical approaches. The resulting distributions are visualized in Fig. 15 (top).

Finally we prove that $g(x \mid A, \mu, \sigma)$ ***is concave for*** x ***in the interval*** $[\mu - \sigma, \mu + \sigma]$***. This means that in the interested region*** δ ***(near the normalized level*** p^l***), its rate of change is slow and increases as*** x ***diverges from*** μ***.***

Proof: Given the function:

$$g(x \mid A, \mu, \sigma) = A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{15}$$

we find the first derivative of g with respect to x :

$$g'(x) = \frac{d}{dx}\left[A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)\right]. \tag{16}$$

Using the chain rule, we have:

$$g'(x) = A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \times \frac{d}{dx}\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]. \tag{17}$$

This gives:

$$g'(x) = -A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \times \frac{(x-\mu)}{\sigma^2}. \tag{18}$$

Next, to find the second derivative, differentiate $g'(x)$ with respect to x :

$$g''(x) = \frac{d}{dx}\left[-A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \times \frac{(x-\mu)}{\sigma^2}\right]. \tag{19}$$

Using product rule and simplifying, the result would be:

$$g''(x) = A \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \times \left[\frac{(x-\mu)^2}{\sigma^4} - \frac{1}{\sigma^2}\right]. \tag{20}$$

When $g''(x) < 0$, we have:

$$x \in [\mu - \sigma, \mu + \sigma]. \tag{21}$$

That proves $g(x \mid A, \mu, \sigma)$ is concave in the δ area, and thus its rate of change is slow and increases as x diverges from μ.

Linear function. Its general formulation is

$$g(x \mid A, \mu, \beta) = A - \beta \cdot |x - \mu|, \tag{22}$$

where $A > 0$ denotes the maximum value of the function, $\mu \in [0, 1]$ is the point where the function is symmetric, and $\beta > 0$ determines the function's slope. To satisfy the requirements of a PDF in Section 4.1, the following must hold:

$$\begin{gathered} \int_0^1 (A - \beta \cdot |x - \mu|)\, dx = 1, \\ A - \beta \cdot |x - \mu| \geq 0, \\ A - \beta \cdot |x - \mu| \leq A - \beta \cdot |p^l - \mu|. \end{gathered} \tag{23}$$

From Eqn. 23, we have:

$$\mu = p^l. \tag{24}$$

In practice, with $x \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ being discrete, the equations become:

$$\begin{gathered} \sum_{x \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}} (A - \beta \cdot |x - p^l|) = 1, \\ A - \beta \cdot |x - p^l| \geq 0. \end{gathered} \tag{25}$$

Given a specific normalized level p^l and varying A, $g(x \mid A, \mu, \beta)$ values are computed for different x. The resulting distributions are visualized in Fig. 15 (middle).

Finally, we prove that $g(x \mid A, \mu, \beta)$ ***has no curvature, and thus its rate of change is consistent regardless of the value of*** x**.**

Proof: Given the function:

$$g(x \mid A, \mu, \beta) = A - \beta \cdot |x - \mu|, \tag{26}$$

we will differentiate this function based on the absolute value, which will result in two cases for the derivatives based on the sign of $(x - \mu)$.

Case 1: $x > \mu$: In this case, $|x - \mu| = x - \mu$. So, $g(x \mid A, \mu, \beta) = A - \beta \cdot (x - \mu)$.

First derivative $g'(x)$:

$$g'(x) = \frac{d}{dx}(A - \beta \cdot (x - \mu)) = -\beta. \tag{27}$$

Second derivative $g''(x)$:

$$g''(x) = \frac{d}{dx}(-\beta) = 0. \tag{28}$$

Case 2: $x < \mu$: In this case, $|x - \mu| = \mu - x$. So, $g(x \mid A, \mu, \beta) = A - \beta \cdot (\mu - x)$.

https://doi.org/10.52202/079017-0460

First derivative $g'(x)$:

$$g'(x) = \frac{d}{dx}(A - \beta \cdot (\mu - x)) = \beta. \tag{29}$$

Second derivative $g''(x)$:

$$g''(x) = \frac{d}{dx}(\beta) = 0. \tag{30}$$

When the second derivative is constantly 0, it means the function has no curvature and its rate of change is constant at every point.

Exponential function. Its general formulation is

$$g(x \mid A, \mu, \lambda) = A \cdot \lambda \cdot \exp(-\lambda \cdot |x - \mu|), \tag{31}$$

where $A > 0$ denotes the intensity of the function, $\mu \in [0, 1]$ is the point where the function is symmetric, and $\lambda > 0$ determines the spread or width of the function. To satisfy the requirements of a PDF in Section 4.1, the following must hold:

$$\begin{gathered} \int_0^1 (A \cdot \lambda \cdot \exp(-\lambda \cdot |x - \mu|))\, dx = 1, \\ A \cdot \lambda \cdot \exp(-\lambda \cdot |x - \mu|) \geq 0, \\ A \cdot \lambda \cdot \exp(-\lambda \cdot |x - \mu|) \leq A \cdot \lambda \cdot \exp(-\lambda \cdot |p^l - \mu|). \end{gathered} \tag{32}$$

From Eqn. 32, we have:

$$\mu \ = \ p^l. \tag{33}$$

In practice, with $x \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ being discrete, the equations become:

$$\begin{gathered} \sum_{x \in \{0, 0.2, 0.4, 0.6, 0.8, 1\}} (A \cdot \lambda \cdot \exp(-\lambda \cdot |x - p^l|)) = 1, \\ A \cdot \lambda \cdot \exp(-\lambda \cdot |x - p^l|) \geq 0. \end{gathered} \tag{34}$$

Given a specific normalized level p^l and varying A, $g(x \mid A, \mu, \lambda)$ values are computed for different x using numerical approaches. The resulting distributions are visualized in Fig. 15 (bottom).

Finally, we prove that $g(x \mid A, \mu, \lambda)$ ***is convex: its rate of change is rapid in the*** δ ***area and decreases as*** x ***moves diverges from*** μ***.***

Proof: Given the function:

$$g(x \mid A, \mu, \lambda) = A \cdot \lambda \cdot \exp(-\lambda \cdot |x - \mu|), \tag{35}$$

we find the first and second derivatives with respect to x. This function involves an absolute value, which will create two cases for the derivatives based on the sign of $(x - \mu)$. Case 1: $x > \mu$: In this case, $|x - \mu| = x - \mu$. So,

$$g(x \mid A, \mu, \lambda) = A \cdot \lambda \cdot \exp(-\lambda \cdot (x - \mu)). \tag{36}$$

First derivative $g'(x)$:

$$\begin{aligned} g'(x) &= A \cdot \lambda \cdot \frac{d}{dx} \exp(-\lambda \cdot (x - \mu)), \\ g'(x) &= -A \cdot \lambda^2 \cdot \exp(-\lambda \cdot (x - \mu)). \end{aligned} \tag{37}$$

Second derivative $g''(x)$:

$$\begin{aligned} g''(x) &= -A \cdot \lambda^2 \cdot \frac{d}{dx} \exp(-\lambda \cdot (x - \mu)), \\ g''(x) &= A \cdot \lambda^3 \cdot \exp(-\lambda \cdot (x - \mu)) > 0. \end{aligned} \tag{38}$$

Case 2: $x < \mu$: In this case, $|x - \mu| = \mu - x$. So,

$$g(x \mid A, \mu, \lambda) = A \cdot \lambda \cdot \exp(-\lambda \cdot (\mu - x)). \tag{39}$$

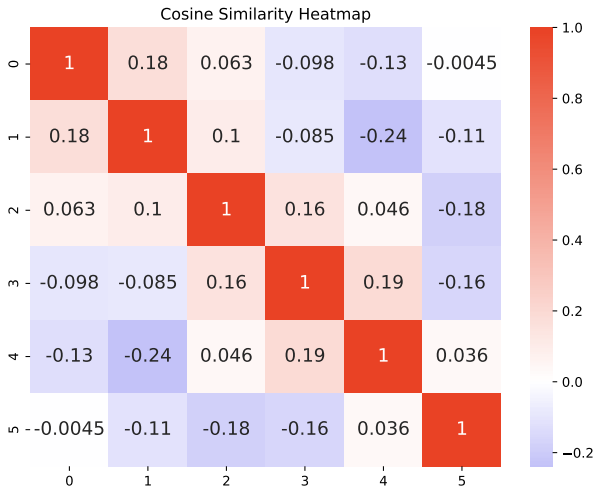

Figure 7: The cosine similarity heatmap of the learned vectors.

First derivative $g'(x)$:

$$\begin{aligned} g'(x) &= A \cdot \lambda \cdot \frac{d}{dx} \exp(-\lambda \cdot (\mu - x)), \\ g'(x) &= A \cdot \lambda^2 \cdot \exp(-\lambda \cdot (\mu - x)). \end{aligned} \tag{40}$$

Second derivative $g''(x)$:

$$\begin{aligned} g''(x) &= A \cdot \lambda^2 \cdot \frac{d}{dx} \exp(-\lambda \cdot (\mu - x)), \\ g''(x) &= A \cdot \lambda^3 \cdot \exp(-\lambda \cdot (\mu - x)) > 0. \end{aligned} \tag{41}$$

When the second derivative is bigger than 0, it means the function is convex: its rate of change is rapid in the δ area and decreases as x moves diverge from μ.

C The Visualization of What the Network Learns

To gain insights into what the network has learned, we offer two visualization methods. First, we present the cosine similarity heatmap of the learned $\mathbf{C}^0 = \left[\mathbf{c}_0^0, \mathbf{c}_1^0, \ldots, \mathbf{c}_N^0\right]$ (refer to Fig. 7). Second, we show the distribution changes of image pairs throughout the training process. The final epoch's distribution can be seen in Fig. 16, while the entire training process is depicted in the attached videos.

From the heatmap, we conclude that: (1) The cosine similarity between different vectors is very low. This demonstrates that the learned vectors are linearly independent. (2) Neighboring vectors exhibit a relatively high cosine similarity. This is consistent with the expectation, as they correspond to similar replication levels.

From the observed changes in the distributions, we note that: (1) While the distribution initially starts as a uniform distribution or peaks at an incorrect level, the network, after training, eventually produces an appropriate and accurate distribution for each image pair. (2) For instance, when supervised by the Gaussian distribution, the network, as expected, produces a final distribution that, though not perfect, closely imitates this.

D Implement GPT-4V Turbo on our D-Rep Test Dataset

This section details implementing GPT-4V Turbo on our D-Rep test dataset. GPT-4V Turbo, which has been online since November 6, 2023, is the latest and most powerful large multimodal model developed by OpenAI. Because it cannot be regarded as a feature extractor, we directly prompt it with two images and one instruction:

https://doi.org/10.52202/079017-0460

Give you one pair of images; please give the similarity of the second one to the first one. Diffusion Models generate the second one, while the first one is the original one. Please only reply with one similarity from $0 - 1$*; no other words are needed. Please understand the images by yourself.*

Given these prompts, GPT-4V Turbo returns a similarity ranging from 0 to 1. Using the official API, we ask the GPT-4V Turbo to determine all similarities between the image pairs in the D-Rep test dataset. Note that the computational cost of employing GPT-4V Turbo in practical applications is prohibitively high. Specifically, to compare an image against an image database containing one million images, the API must be called one million times, incurring a cost of approximately $\$7,800$.

E The Similarities Predicted by Other Models

In Fig. 17, we show the similarities predicted by six selected models (two vision-language models, two current ICD models, and two others). We conclude that: (1) CLIP [32] tends to assign higher similarities, which deteriorates its performance on image pairs with low levels, leading to many false positive predictions; (2) GPT-4V Turbo [44] and DINOv2 [49] can produce both high and low predictions, but its performance does not match ours. (3) The prediction ranges of ResNet-50 [53] are relatively narrow, indicating its inability to distinguish image pairs with varying levels effectively. (4) Current ICD models, including SSCD [15] and BoT [17], consistently produce low predictions. This is because they are trained for invariance to image transformations (resulting in high similarities for pirated content produced by transformations) and cannot handle replication generated by diffusion models.

F The Details of Six Diffusion Models

This section provides details on the evaluation sources for three commercial and three open-source diffusion models.

Midjourney [3] was developed by the Midjourney AI company. We utilized a dataset scraped by Succinctly AI under the cc0-1.0 license. This dataset comprises $249,734$ images generated by real users from a public Discord server.

New Bing [4], also known as Bing Image Creator, represents Microsoft's latest text-to-image technique. We utilized the repository under the Unlicense to generate $216,957$ images. These images were produced using randomly selected prompts from DiffusionDB [13].

DALLE·2 [5] is a creation of OpenAI. We downloaded all generated images directly from this website, resulting in a dataset containing $50,904$ images. We have obtained permission from the website's owners to use the images for research purposes.

We downloaded and deployed three open-source diffusion models, including **SDXL** [6], **DeepFloyd IF** [7], and **GLIDE** [8]. These models were set up on a local server equipped with 8 A100 GPUs. Distributing on them, we generated $1,819,792$ images with prompts from DiffusionDB [13].

G More Replication Examples

We provide more replication examples by diffusion models in Fig. 18 to Fig. 23.

H Failure Cases

As shown in Fig. 8, we identify two primary failure cases. The first type of failure occurs when the generated images replicate only common elements without constituting replicated content. For instance, elements like grass (Midjourney), helmets (New Bing), and buildings (DALL·E 2) appear frequently but do not indicate actual replication of content. The second type of failure arises when two images share high-level semantic similarity despite having no replicated content. An example can be seen in the image pairs where themes, styles, or concepts are similar, such as the presence of iconic structures (SDXL and GLIDE) or stylized portraits (DeepFloyd IF), even if the specific

Figure 8: The failure cases of our detection method. We show one example for each diffusion model.

content is not replicated. Understanding these failure modes is crucial for improving the accuracy and robustness of our detection methods in the future.

https://doi.org/10.52202/079017-0460

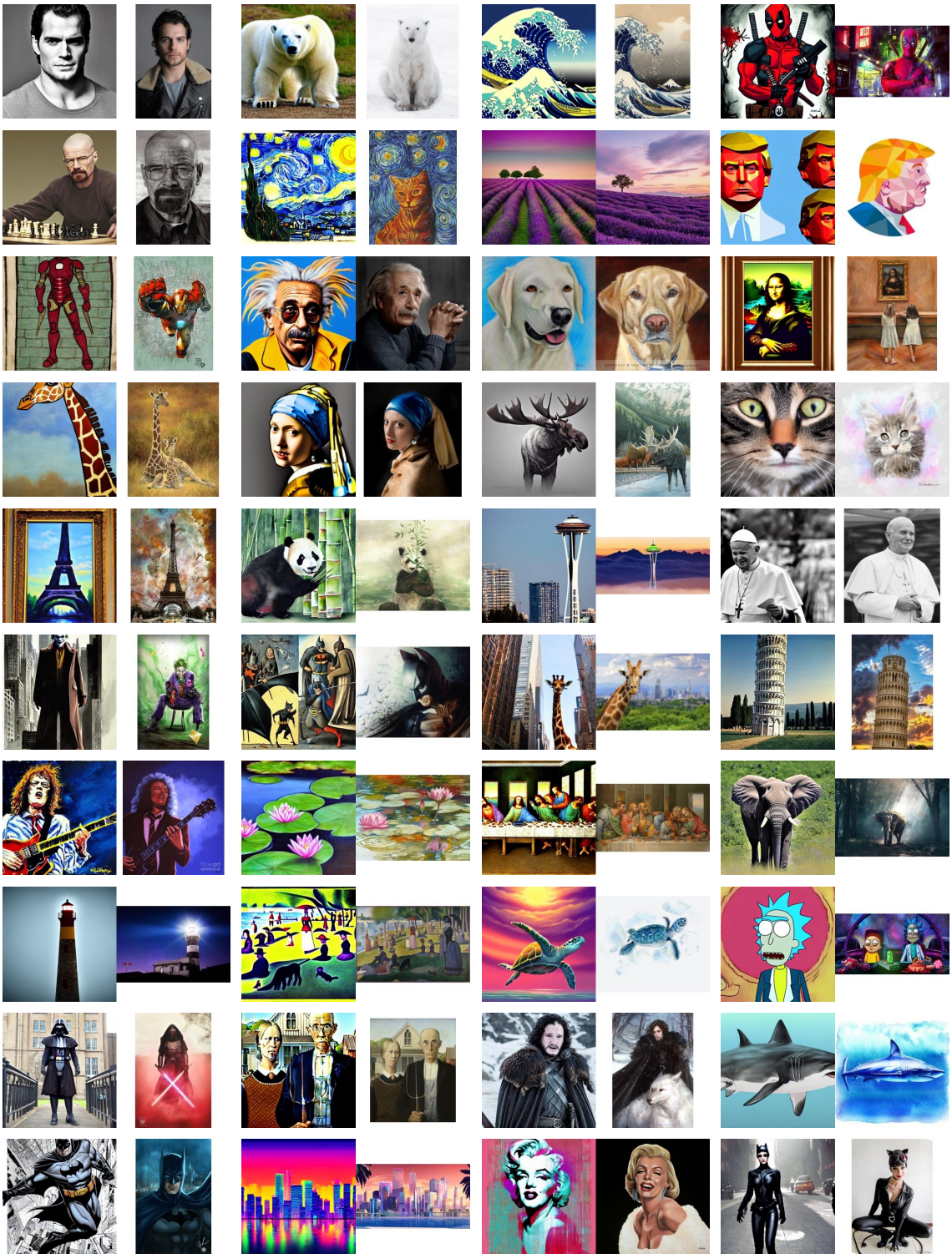

Figure 9: The example image pairs with level 5.

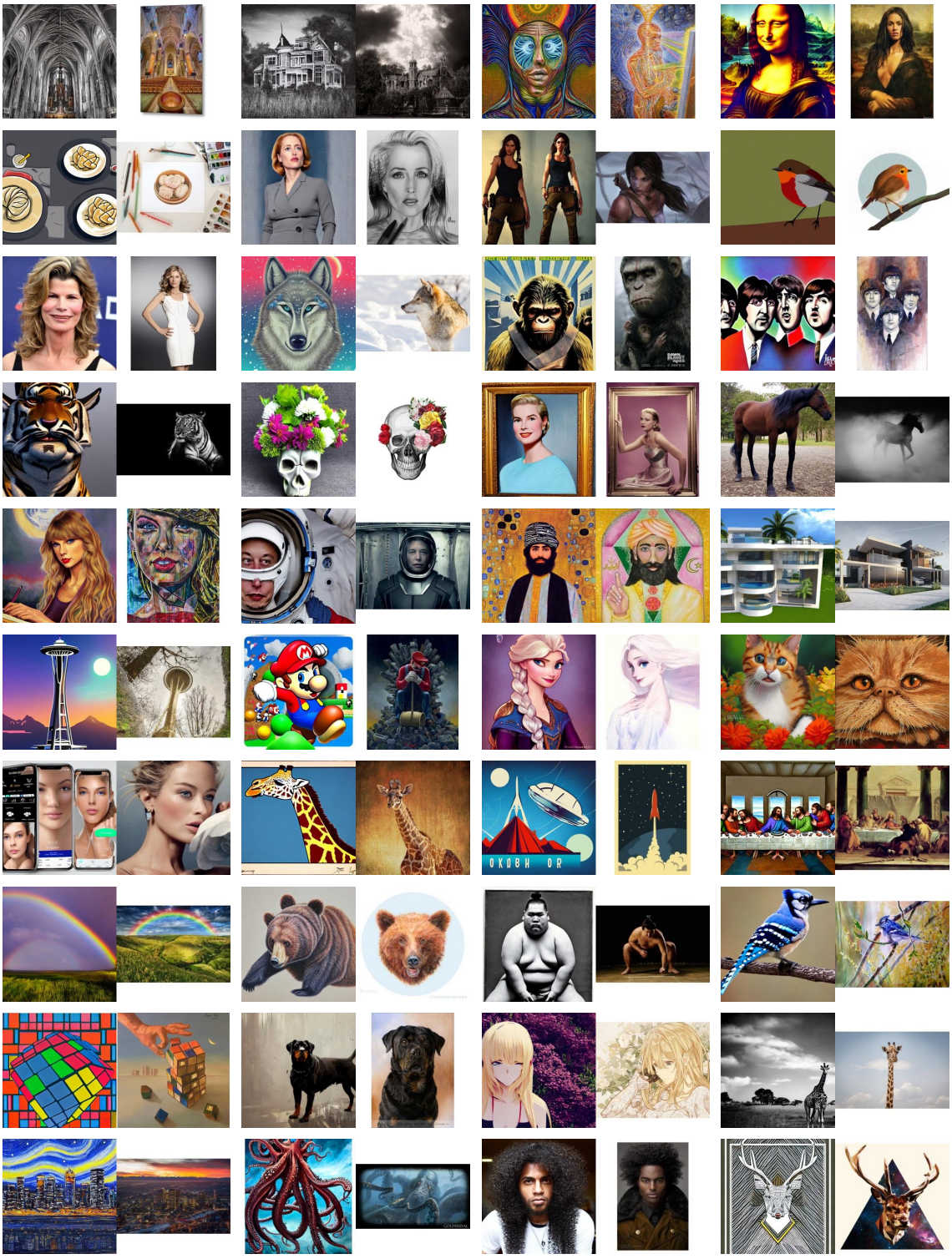

Figure 10: The example image pairs with level 4.

https://doi.org/10.52202/079017-0460

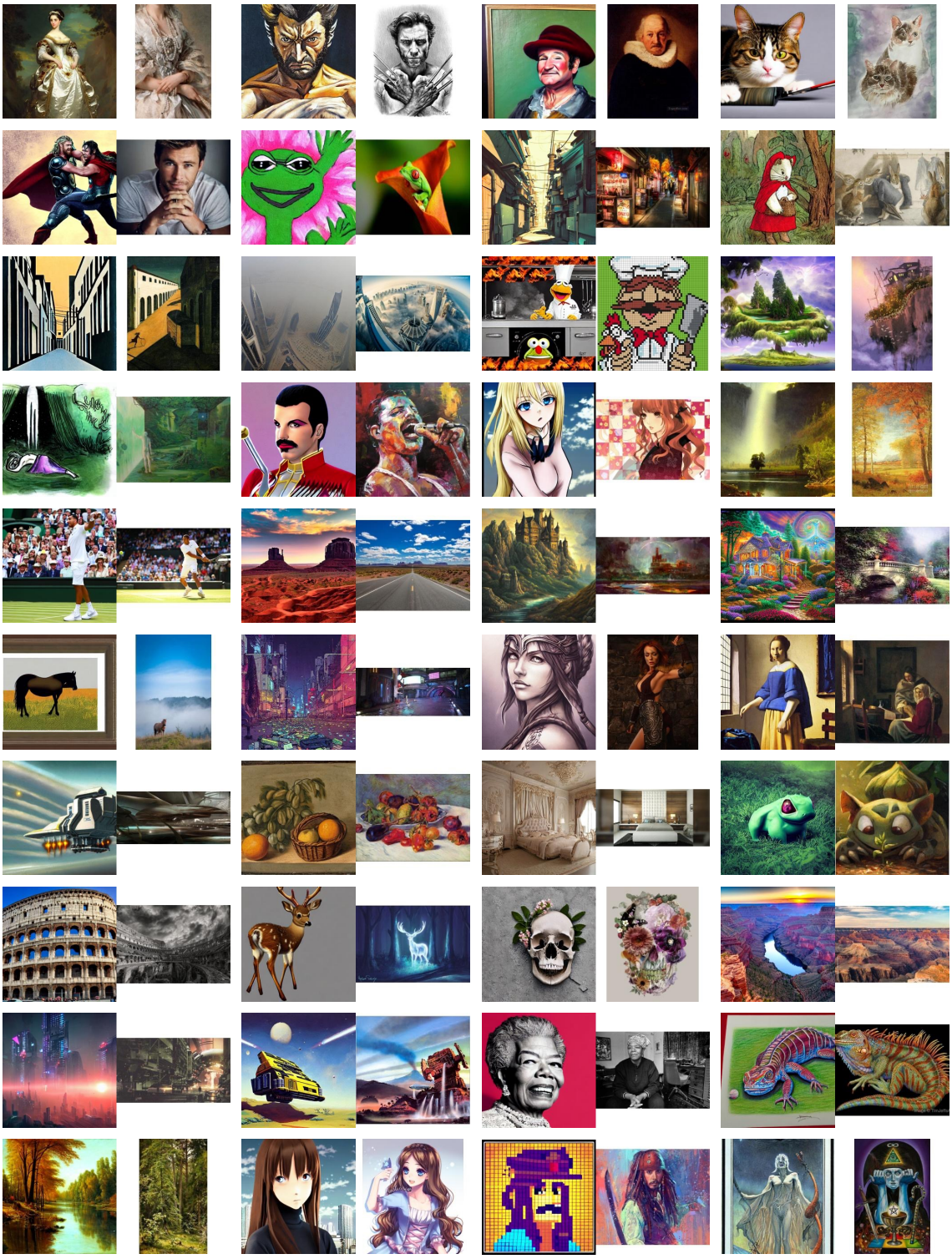

Figure 11: The example image pairs with level 3.

https://doi.org/10.52202/079017-0460

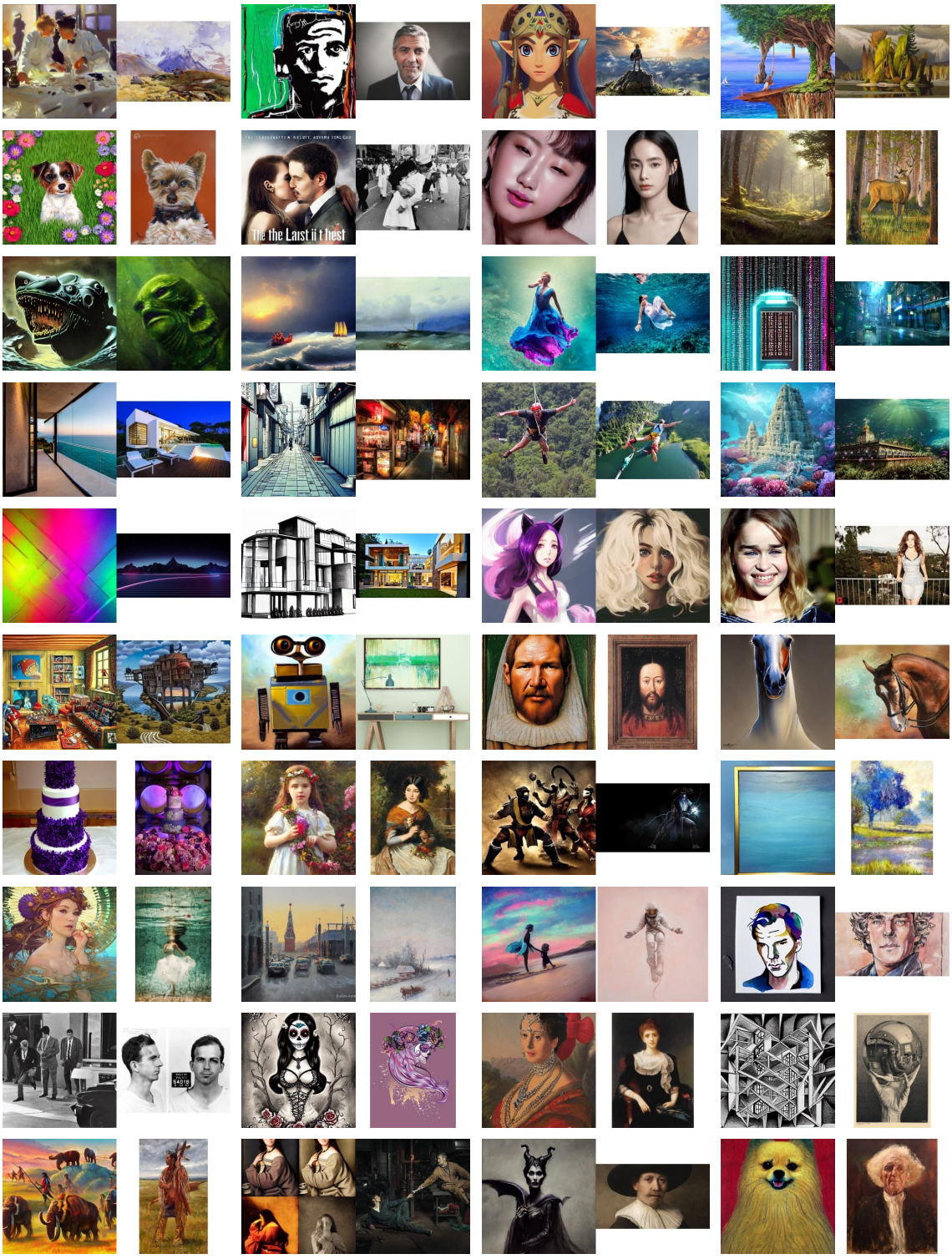

Figure 12: The example image pairs with level 2.

https://doi.org/10.52202/079017-0460

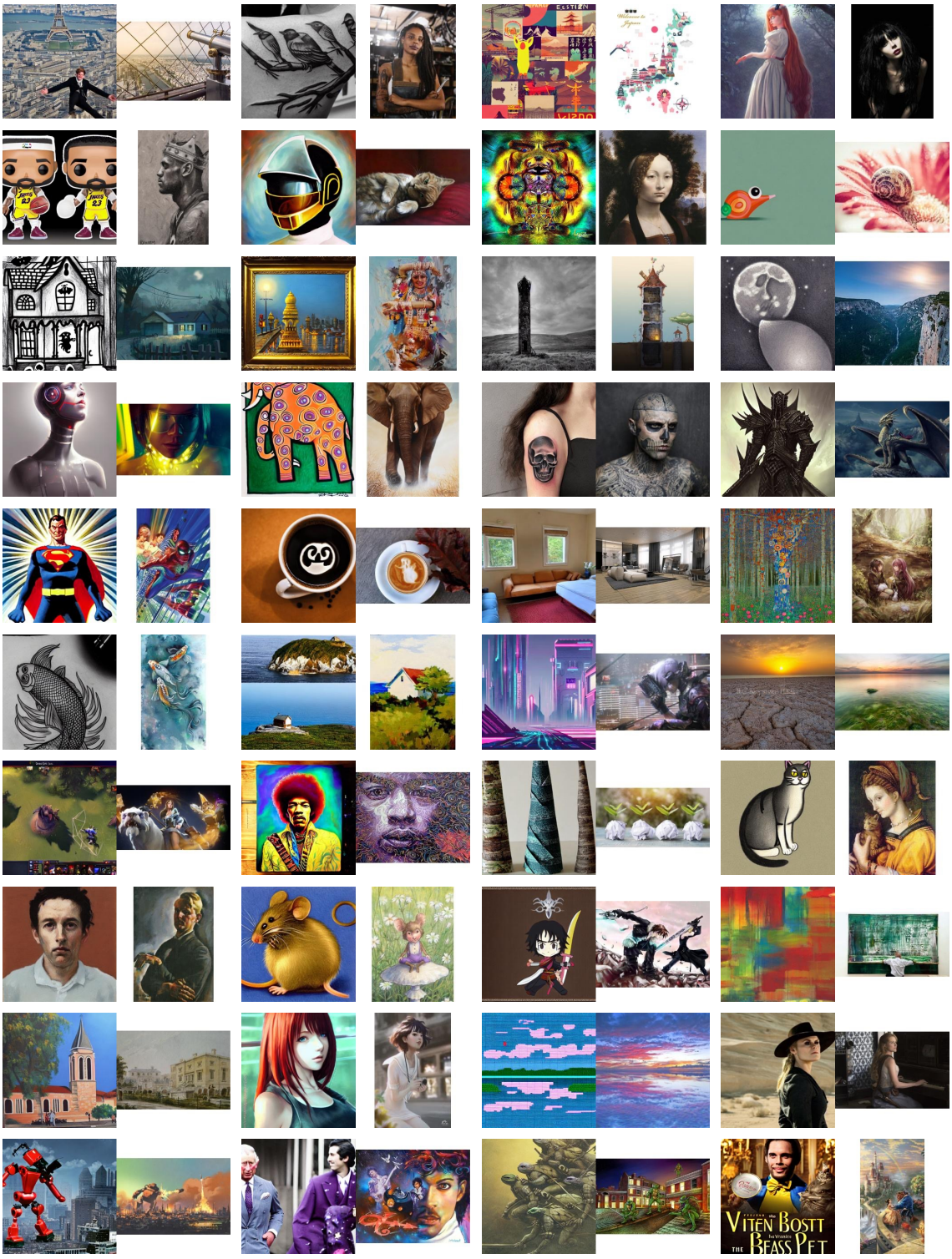

Figure 13: The example image pairs with level 1.

https://doi.org/10.52202/079017-0460

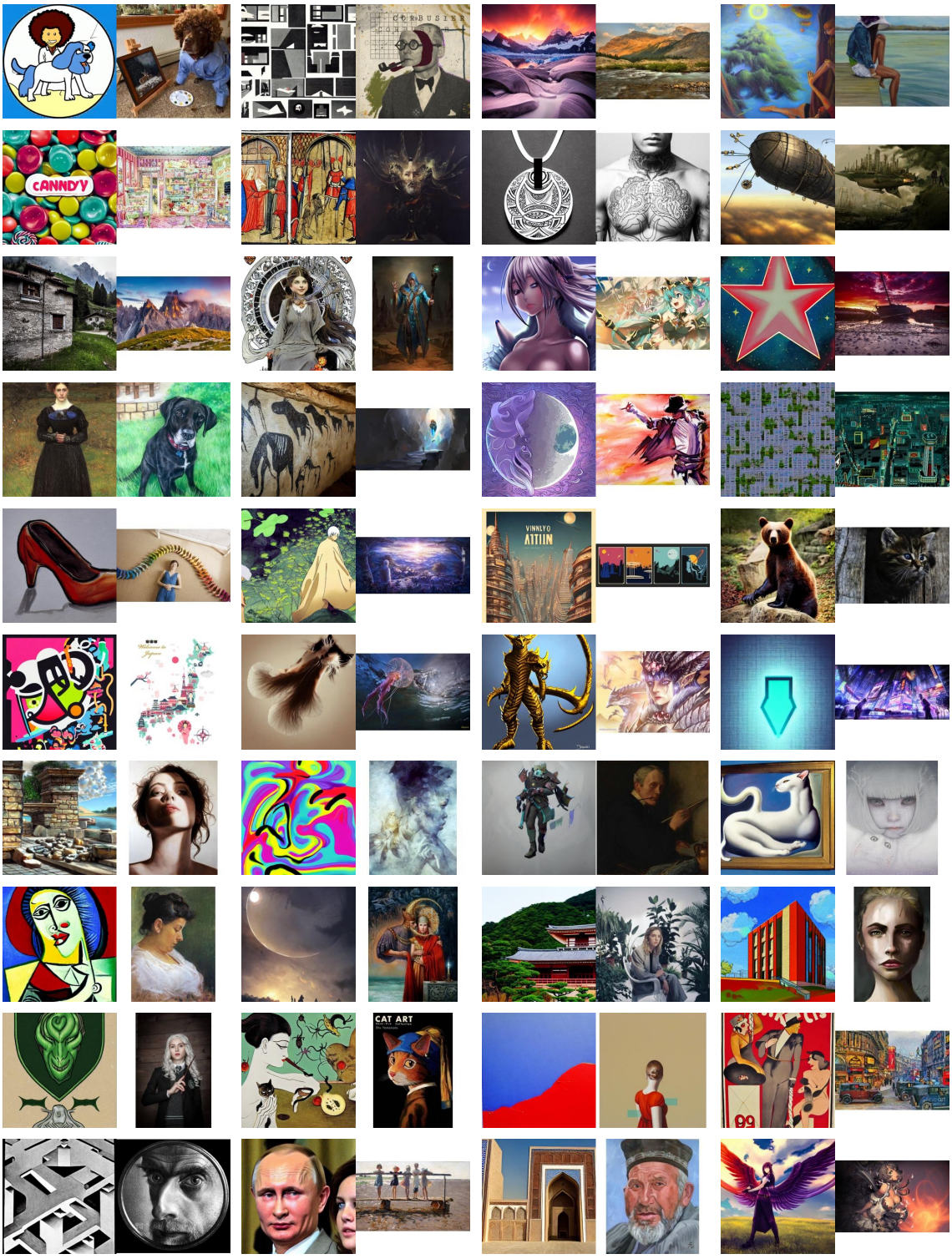

Figure 14: The example image pairs with level 0.

https://doi.org/10.52202/079017-0460

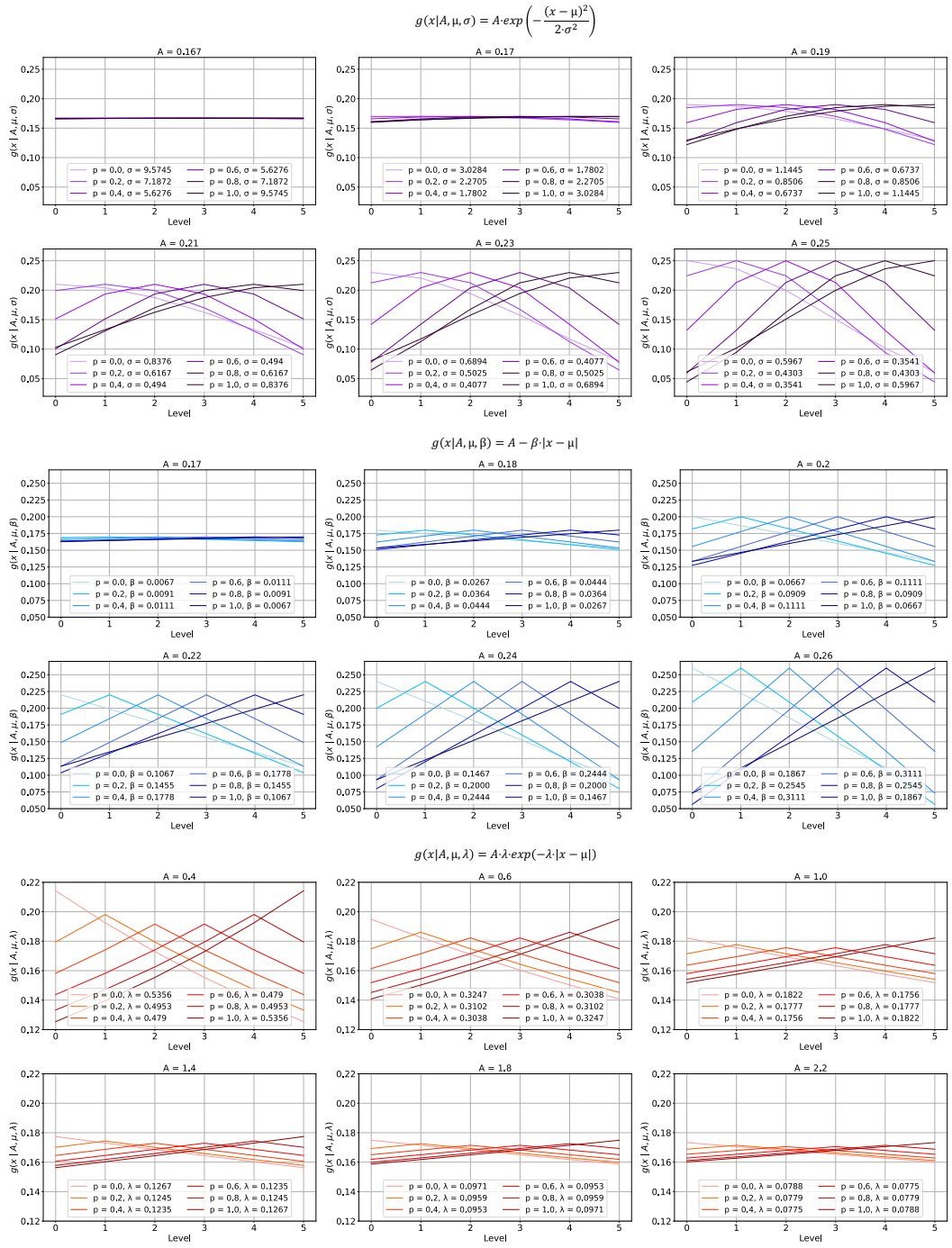

Figure 15: The distributions converted from replication levels. We use Gaussian, linear, and exponential functions as the representative demonstrations.

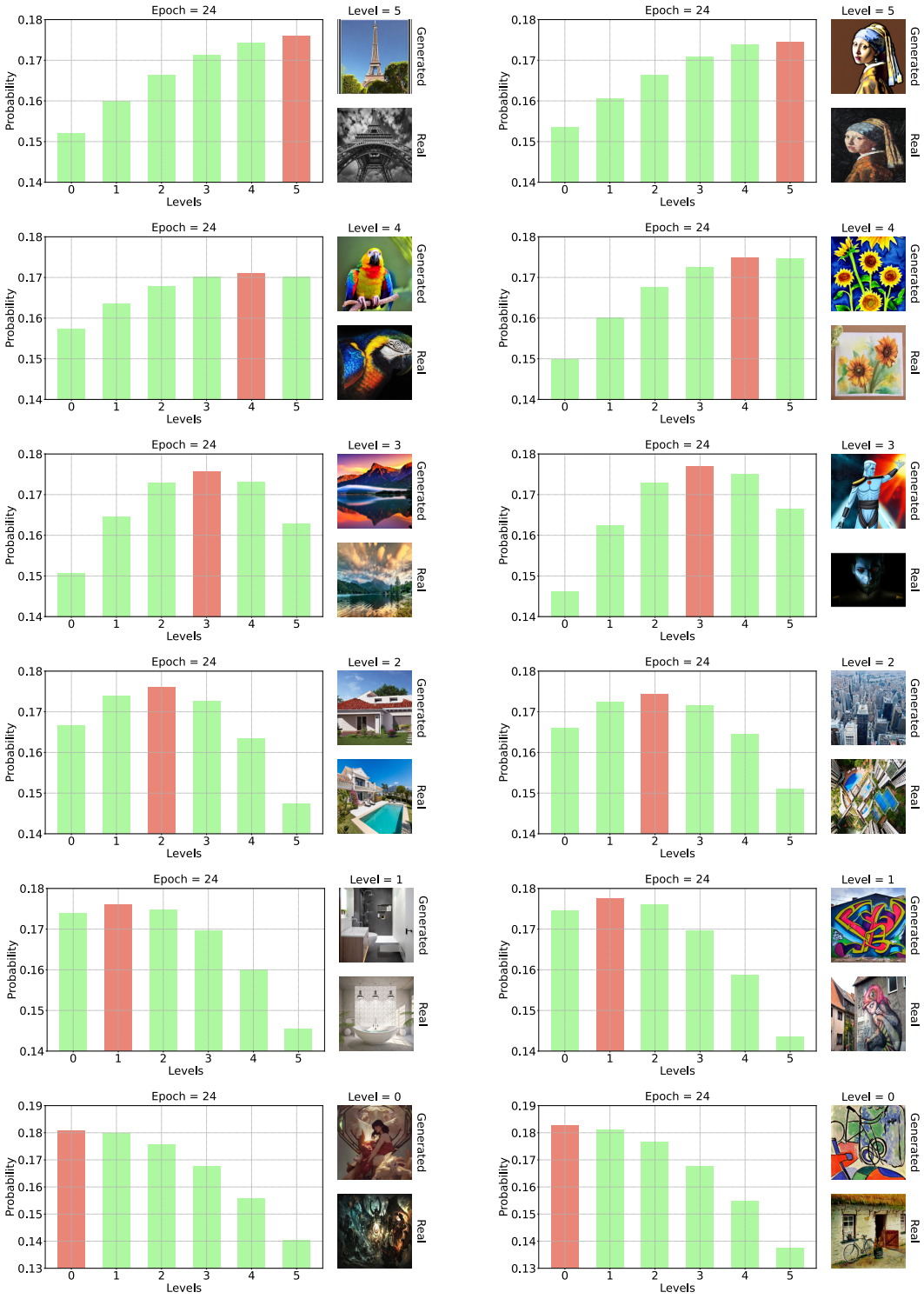

Figure 16: The learned distributions of different image pairs. Please see the attached videos for the distribution changes in the whole training process.

https://doi.org/10.52202/079017-0460

Method		Sim.		Sim.		Sim.		Sim.
CLIP		0.63		0.68		0.85		0.87
GPT-4V		0.40		0.60		0.30		0.90
DINOv2		0.35		0.65		0.69		0.35
ResNet-50		0.54		0.61		0.53		0.58
SSCD		0.04		0.16		0.08		0.10
BoT		0.05		0.36		0.21		0.06
Label		1.00		1.00		1.00		1.00

Method		Sim.		Sim.		Sim.		Sim.
CLIP		0.49		0.57		0.65		0.68
GPT-4V		0.70		0.20		0.00		0.30
DINOv2		0.40		0.60		0.21		0.44
ResNet-50		0.67		0.56		0.65		0.56
SSCD		0.11		0.13		0.00		0.12
BoT		0.24		0.36		0.11		0.25
Label		0.80		0.80		0.80		0.80

Method		Sim.		Sim.		Sim.		Sim.
CLIP		0.70		0.71		0.82		0.93
GPT-4V		0.70		0.20		0.30		0.80
DINOv2		0.61		0.59		0.40		0.76
ResNet-50		0.60		0.60		0.57		0.62
SSCD		0.08		0.02		0.08		0.31
BoT		0.25		0.03		0.22		0.35
Label		0.60		0.60		0.60		0.60

Method		Sim.		Sim.		Sim.		Sim.
CLIP		0.57		0.57		0.62		0.77
GPT-4V		0.20		0.30		0.10		0.30
DINOv2		0.28		0.58		0.23		0.55
ResNet-50		0.67		0.60		0.63		0.66
SSCD		0.01		0.14		0.15		0.10
BoT		0.21		0.21		0.25		0.16
Label		0.40		0.40		0.40		0.40

Method		Sim.		Sim.		Sim.		Sim.
CLIP		0.39		0.55		0.56		0.69
GPT-4V		0.00		0.00		0.00		0.00
DINOv2		0.07		0.17		0.11		0.11
ResNet-50		0.59		0.59		0.59		0.56
SSCD		0.06		0.02		0.05		0.12
BoT		0.15		0.01		0.05		0.05
Label		0.20		0.20		0.20		0.20

Method		Sim.		Sim.		Sim.		Sim.
CLIP		0.38		0.56		0.57		0.78
GPT-4V		0.00		0.00		0.00		0.10
DINOv2		0.06		0.02		0.09		0.38
ResNet-50		0.61		0.62		0.53		0.63
SSCD		0.05		0.05		0.04		0.23
BoT		0.12		0.08		0.03		0.43
Label		0.00		0.00		0.00		0.00

Figure 17: The similarities (or normalized levels) predicted by existing models.

https://doi.org/10.52202/079017-0460

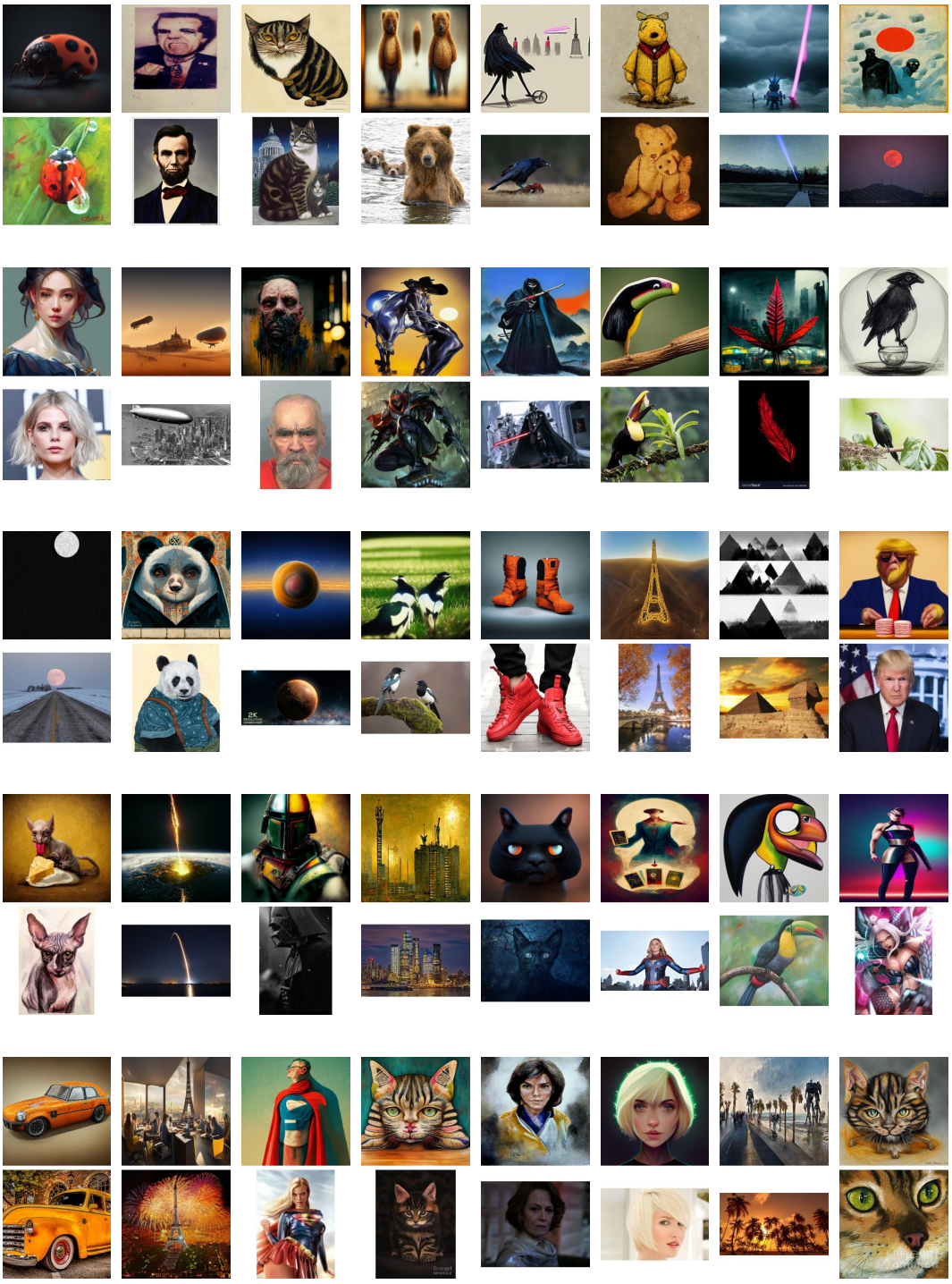

Figure 18: The replication examples generated by Midjourney [3].

https://doi.org/10.52202/079017-0460

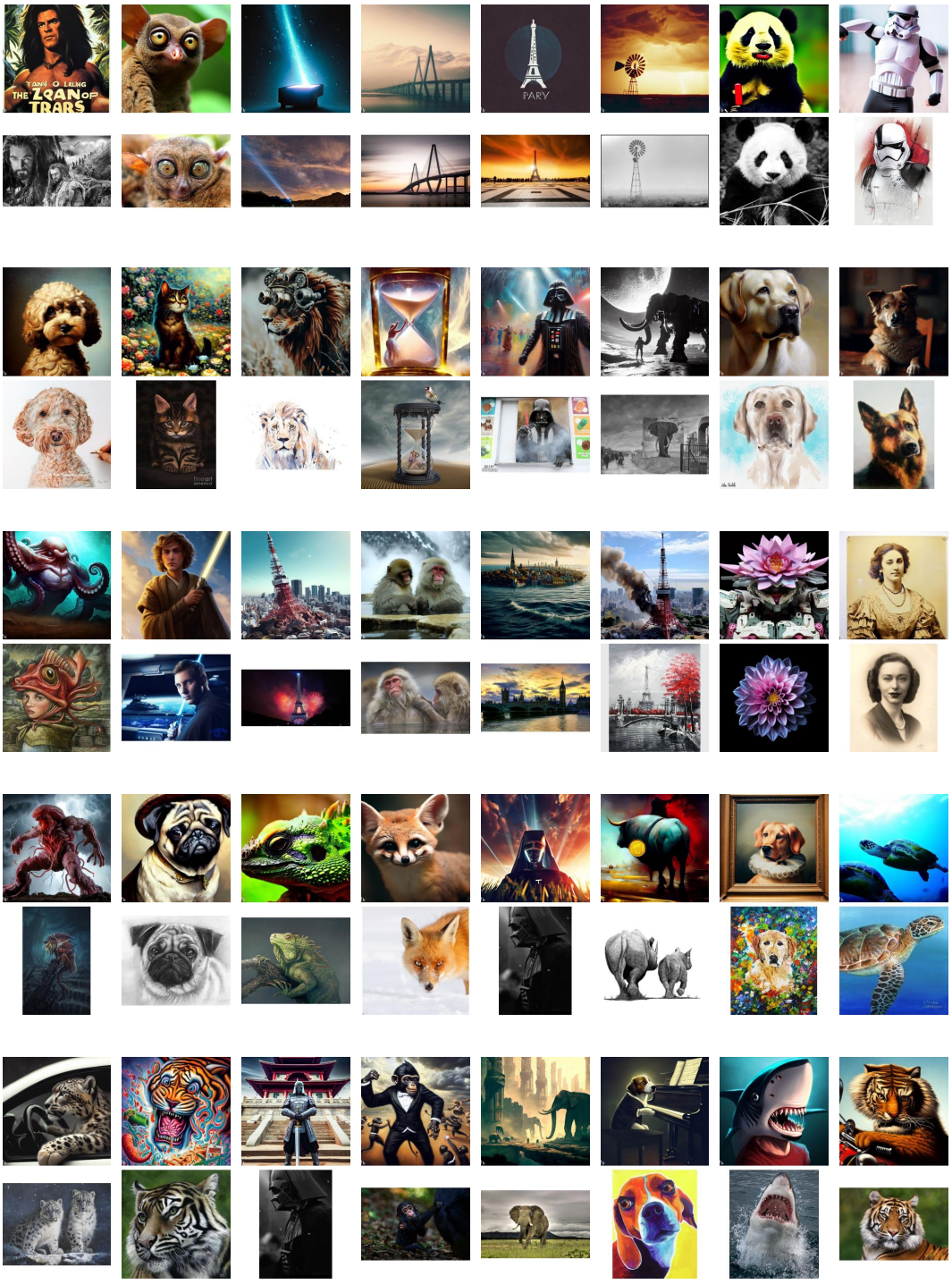

Figure 19: The replication examples generated by New Bing [4].

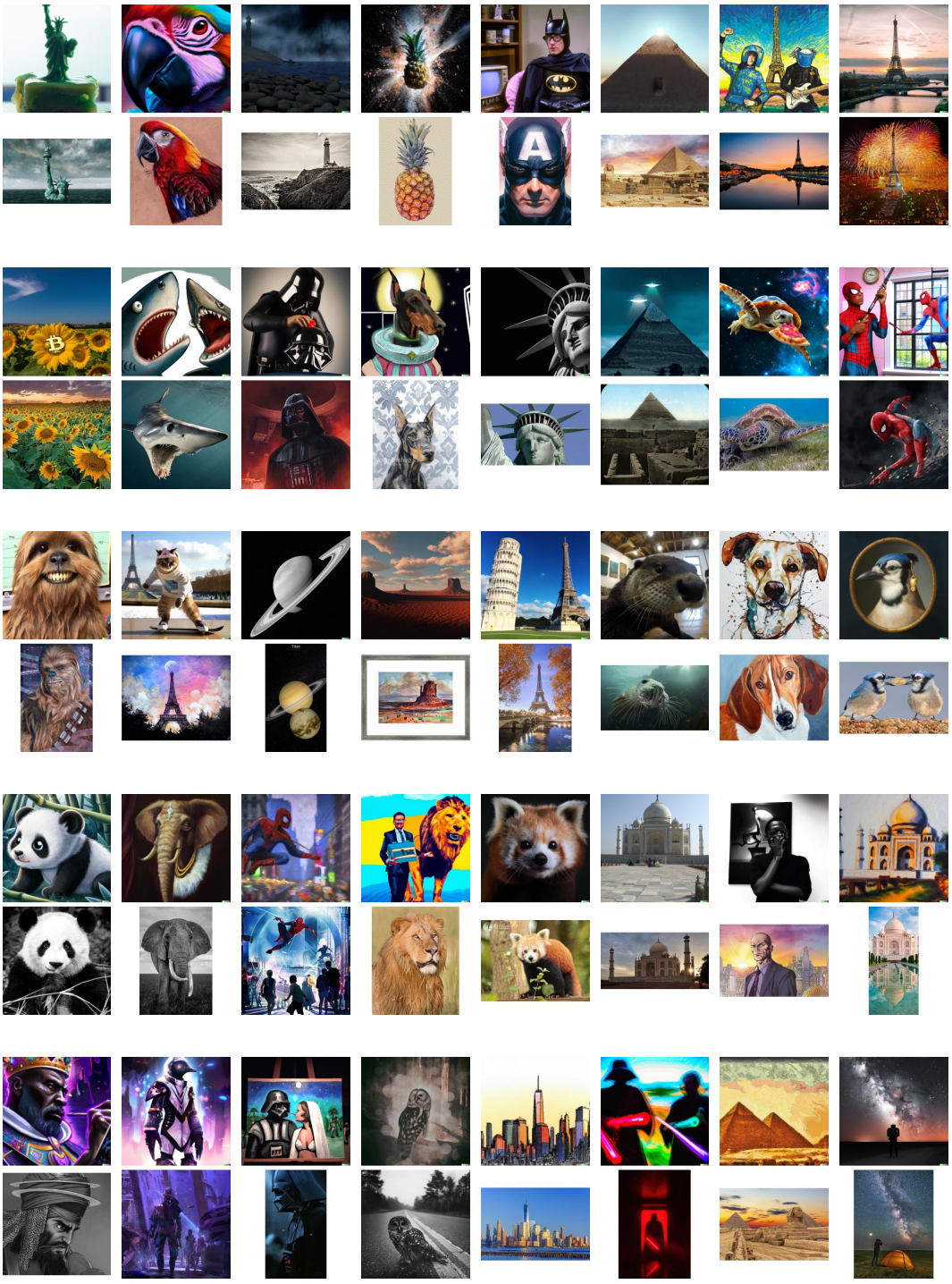

Figure 20: The replication examples generated by DALLE·2 [5].

https://doi.org/10.52202/079017-0460

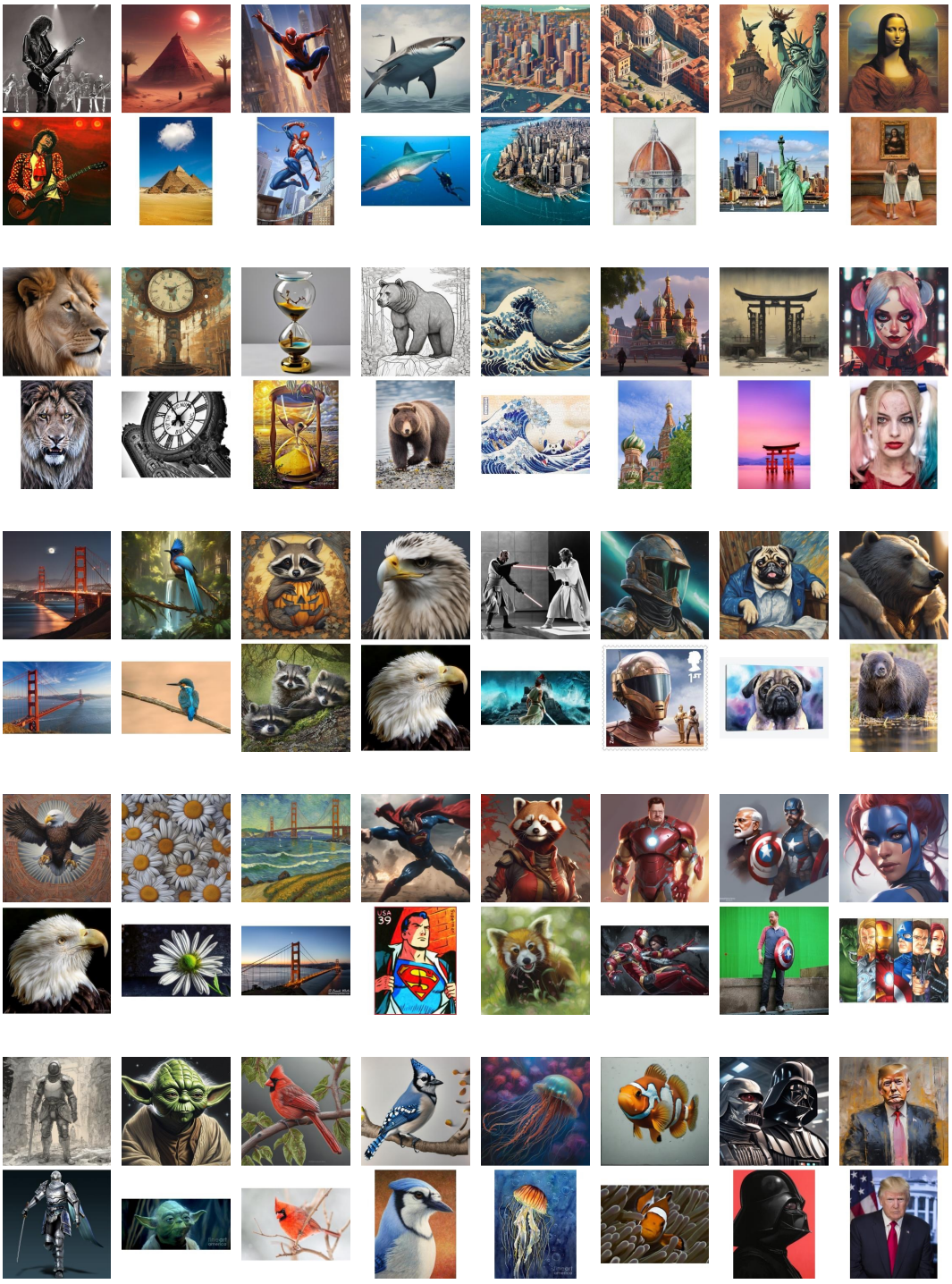

Figure 21: The replication examples generated by SDXL [6].

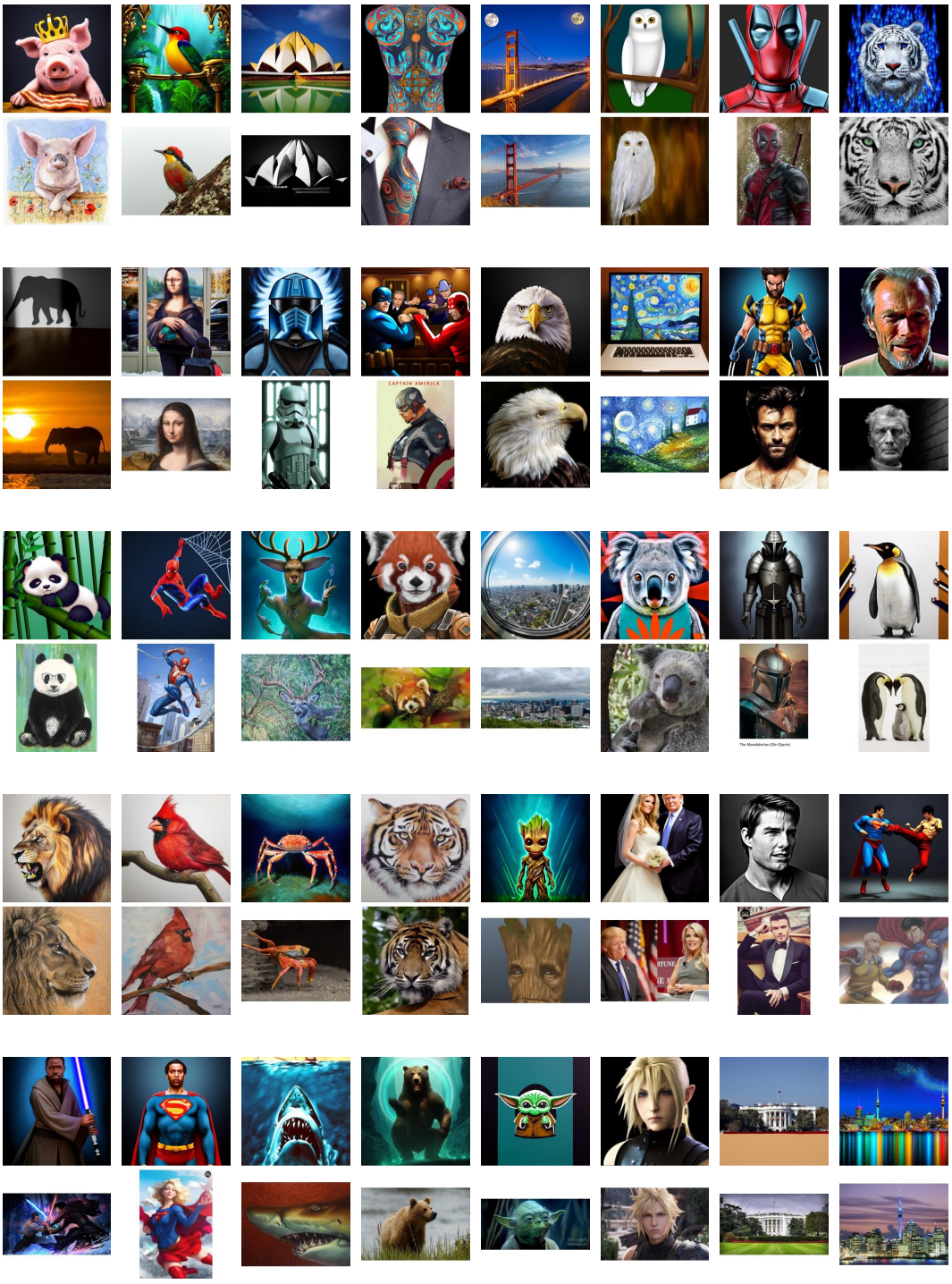

Figure 22: The replication examples generated by DeepFloyd IF [7].

https://doi.org/10.52202/079017-0460

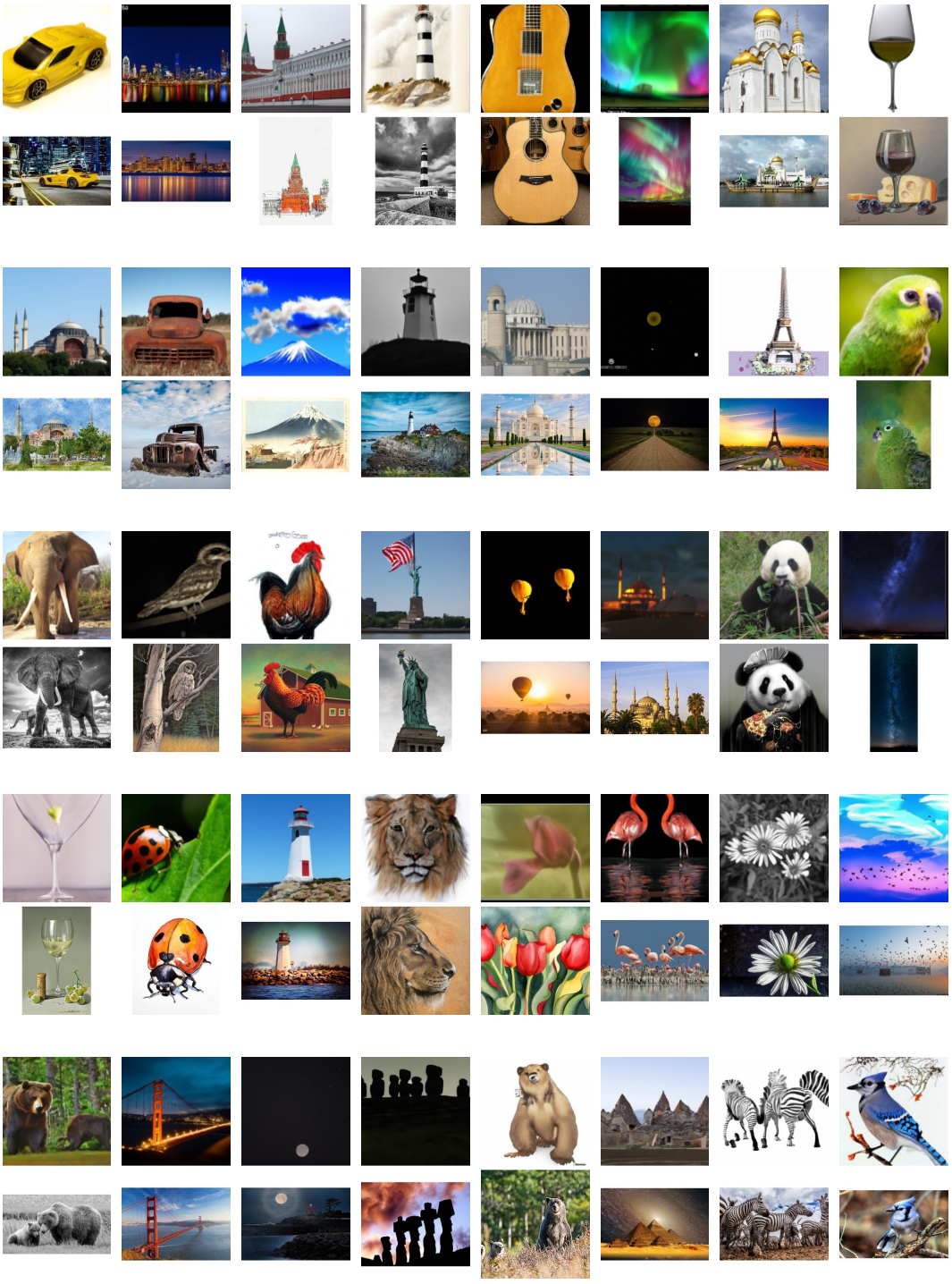

Figure 23: The replication examples generated by GLIDE [8].

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We follow this common practice for writing a paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: See the Disclaimer at the end of our main paper.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

https://doi.org/10.52202/079017-0460

Justification: This is not a theory paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the detail of our deigned model in the main paper and will release the code upon the acceptance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The data and code are publicly available at https://icdiff.github.io/.

Guidelines:

- The answer NA means that paper does nfot include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental settings and details are provided in the main paper as well as the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: All the experimental results are significant and stable, and error bars are not reported because it would be too computationally expensive.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-0460

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include the experiments compute resources in the section of Experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please see the end of Conclusion and the Disclaimer.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We cite all the assets used in our paper, and all the corresponding licenses are mentioned and respected.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-0460

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The data and code are publicly available at https://icdiff.github.io/.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [Yes]

    Justification: We only involve crowdsourcing to curate our dataset. The salaries of workers in the data curation process are paid beyond the minimum wage. Our research has nothing to do with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: We do not involve research with human subjects and there is no potential risk.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.