Method for Exploring Generative Adversarial Networks (GANs) via Automatically Generated Image Galleries

Enhao Zhang
ehzhang@umich.edu
University of Michigan
Ann Arbor, Michigan

Nikola Banovic
nbanovic@umich.edu
University of Michigan
Ann Arbor, Michigan

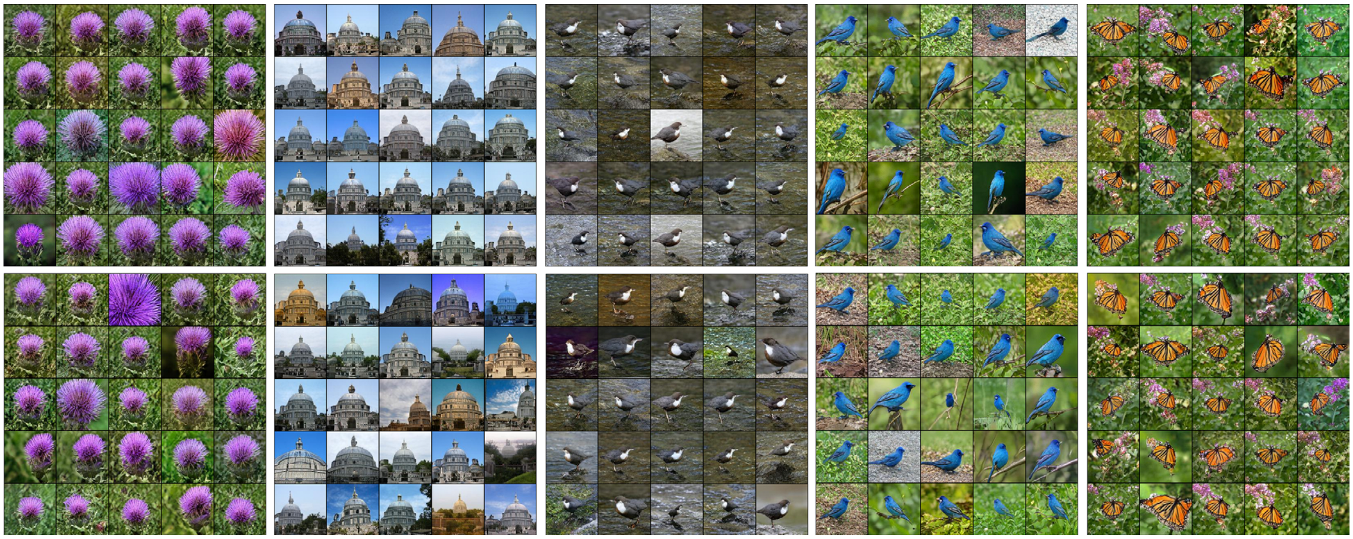


Figure 1: A selection of image galleries generated from the BigGAN model [6] using our interactive GAN exploration interface (top row) and automatically sampled from the GAN model using our sampling method (bottom row).

ABSTRACT

Generative Adversarial Networks (GANs) can automatically generate quality images from learned model parameters. However, it remains challenging to explore and objectively assess the quality of all possible images generated using a GAN. Currently, model creators evaluate their GANs via tedious visual examination of generated images sampled from narrow prior probability distributions on model parameters. Here, we introduce an interactive method to explore and sample quality images from GANs. Our first two user studies showed that participants can use the tool to explore a GAN and select quality images. Our third user study showed that images sampled from a posterior probability distribution using a Markov Chain Monte Carlo (MCMC) method on parameters of images collected in our first study resulted in on average higher quality and more diverse images than existing baselines. Our work enables principled qualitative GAN exploration and evaluation.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CHI '21, May 8–13, 2021, Yokohama, Japan

© 2021 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ACM ISBN 978-1-4503-8096-6/21/05…$15.00
https://doi.org/10.1145/3411764.3445714

CCS CONCEPTS

• **Human-centered computing** → **Interactive systems and tools**.

KEYWORDS

Interactive model exploration, qualitative model validation.

ACM Reference Format:
Enhao Zhang and Nikola Banovic. 2021. Method for Exploring Generative Adversarial Networks (GANs) via Automatically Generated Image Galleries. In *CHI Conference on Human Factors in Computing Systems (CHI '21), May 8–13, 2021, Yokohama, Japan.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3411764.3445714

1 INTRODUCTION

Generative Adversarial Networks (GANs) [14] have shown promise as creativity support tools to automatically generate never before seen images from learned model parameters. In addition to generating images, GANs have been used anywhere from supporting image transformation (e.g., image translation [17, 19], image blending [33] and synthesis [3, 8, 25]) to art tools [15, 20, 29]. As such, they can potentially make people "more creative more often" [31].

However, it remains challenging to explore and objectively assess the ability of a GAN model to generate diverse, high-quality images, where definition of quality varies based on the intended use of the GAN model (e.g., to generate a diverse set of photo-realistic images

[34]). Unlike other generative models that optimize a likelihood function, GANs lack such an objective function, which makes it difficult to quantitatively compare performance of different models [27]. Although there are several quantitative measures to evaluate GANs (e.g., the inception score [27]), it remains unclear which measure best captures capabilities and limitations of a GAN [4]. Furthermore, it is not clear that such quantitative measures capture how humans perceive and judge the quality of generated images.

Thus, visual examination of images by humans remains one of the most common ways to evaluate GANs [4]. Yet, such evaluation [6, 26] involves tedious visual examination of GAN generated images organized into non-interactive image galleries sampled from narrow prior probability distributions on model parameters. Techniques for interactive GAN image generation [3, 10, 13, 18, 30, 35] enable manual creation of specific images, but not sampling of diverse, high-quality images from a GAN. Although recent interactive generative model parameter optimization methods [21, 22] could be used to explore a GAN, they focus on finding a single "best" quality GAN-generated image [22] or a gallery of *similar* high-quality images [21], and not a *diverse* gallery of images required for qualitative GAN exploration and evaluation.

In this paper, we introduce an interactive method for exploring and sampling quality images from GANs to automatically generate galleries of diverse, high-quality images (Figure 1). We first present an interactive tool for exploring GANs and selecting quality images, including their corresponding model parameters (Figure 2). We then show how to use images and model parameters selected using our tool to sample other diverse, high-quality images from a posterior probability distribution of model parameters using a Markov Chain Monte Carlo (MCMC) method [9].

We illustrate our method on the BigGAN model [6], which tests the boundaries of scalability and capabilities of GANs to generate diverse set of photo-realistic images from 1,000 different categories (e.g., Irish Setter, butterfly, dome). We chose BigGAN [6] because it is an exceptionally large model that produces both good and poor quality images. We then evaluated our method in a series of crowdsourced user studies to compare with a current state-of-the-art baseline evaluation method [6], which used non-interactive galleries of randomly sampled images to visually examine a GAN.

We first conducted an Amazon Mechanical Turk (MTurk) [1] user study with 367 participants in which we showed how they used our tool to explore the model and select 10,026 photo-realistic images from ten different BigGAN categories. To validate the output of our first study, we conducted another MTurk user study with 1,622 participants in which they rated 79.94% of images that participants generated using our tool in our first study as photo-realistic. We then conducted our third MTurk user study with 1,000 participants and showed that our method generated more diverse photo-realistic image galleries than the baseline method [6].

Our work enables principled qualitative GAN exploration *via* interactive visual examination, even in regions of the model where one would not expect them to be. Our automated method enables quick generation of diverse, high-quality image galleries to support qualitative evaluation of GANs. Our tools allow users to discover capabilities and limitations of a GAN model through carefully crafted set of interactions with the model. Knowledge we generated in this work will inform future interactive model exploration.

2 GAN EXPLORATION CHALLENGES

Here, we briefly introduce Generative Adversarial Networks (GANs) [14] and explain their underlying architecture before diving deeper into existing methods for exploring and evaluating them. A GAN is a type of generative model, most commonly used to generate images. However, GAN training is unlike training of other discriminative and some generative models that optimize a likelihood function because GANs training does not involve an objective function.

Instead, training a GAN involves two networks: 1) a Generator that takes in a vector of latent variables $\mathbf{z} = (z_1, ..., z_n) \in \mathbb{R}^n$ and outputs the corresponding image, and 2) a Discriminator that is used to distinguish between generated images and real images (provided as training data). The Generator is trained to maximize the probability of fooling the Discriminator, while the Discriminator is trained to discriminate training data from the images created by the Generator. Once trained, the Generator can take any vector of latent variables $\mathbf{z}$ to generate a new image. Our focus is primarily on the Generator and its ability to generate quality images.

This unique property of GANs, where they lack a function to optimize, makes quantitative evaluation of GANs exceptionally difficult [4]. Even current state-of-the-art quantitative methods depend on validation from humans to ensure they capture the human notion of quality [27]. Thus, visual examination and qualitative evaluation still remain an important aspect of GAN evaluation.

However, existing, commonly used qualitative methods use tedious visual examination of image galleries sampled from narrow probability distribution of the image parameters. This can lead to two major challenges: 1) if $\mathbf{z}$ deviates from the mean too greatly, the generated image will tend to have poor quality, and 2) if all elements of $\mathbf{z}$ are too close to the mean, then the resulting images will all look similar and the sample will not be diverse. To account for this, the existing sampling methods [6, 26] sample $\mathbf{z}$ from a truncated normal distribution, with an arbitrarily selected threshold that is the same for all z_i in $\mathbf{z}$. Unfortunately, there is no guarantees that such arbitrary thresholds will not discard some high-quality images from the sample or include poor-quality images in it.

Recent research [3, 10, 13, 18, 30, 35] has proposed a number of techniques for interactive GAN image generation, which all could aid in GAN exploration. Such algorithmic approaches [13, 18] often search specific parts of latent space of a GAN using limited number of interactions with the GAN (e.g., randomly regenerating non-interactive image galleries) to guess how model parameters map to model outputs. Although approaches for direct manipulation of model parameters [30] exist, such interactions currently require tedious manipulation of sliders that map directly to model parameters. Unfortunately, optimization methods [21, 22] that could reduce the time it takes to search model parameters by direct manipulation are not meant for generating galleries of *diverse* images required for qualitative GAN exploration and evaluation.

Direct manipulation of model outputs could increases the expressiveness of interactions with a GAN, but the existing techniques [3, 10, 35] focus on manually generating specific images from a GAN and not a large sample of images that could provide insights into the capabilities and limitations of current GAN models. Thus, it remains unclear what interactions could support interactive GAN exploration and validation.

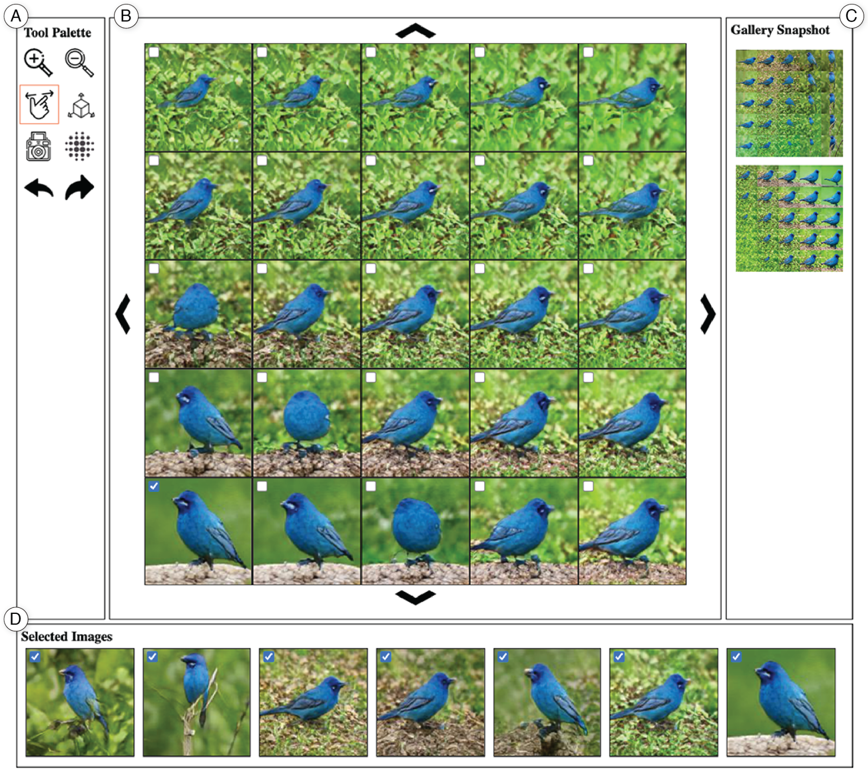


Figure 2: The main components of our final interactive interface for GAN exploration: a) tool palette, b) current working image gallery, c) gallery snapshots, and d) user-selected quality images.

3 INTERACTIVE GAN EXPLORATION

Here, we describe our two-step method for exploring GANs that can aid in qualitative evaluation of a GAN model. In the first step, our method enables exploration of a GAN model through interactive visual examination of a subset of images from a large space of possible images (of both high and poor quality) that the GAN model can generate. In the second step, we show how to use those user-selected quality images from our first step to automatically sample more quality images from the GAN.

3.1 Interactive GAN Exploration Interface

Here we present the design and implementation of an interactive interface (Figure 2) for exploring and selecting high-quality images from GANs. The main design goal of our interface was to enable users to explore the massive space of possible images generated using a GAN in a principled way to find and select quality images.

3.1.1 Iterative Interface Design and Formative Usability User Study. To design and implement our interface, we used an iterative user-centered design approach. We first studied the current context of use and existing GAN exploration and validation methods through existing literature (in particular comprehensive review in [4] and discussion of user needs in [27]), and identified a central user need: the ability to interactively explore a GAN.

We designed an initial interface (Figure 3) for interactive GAN exploration inspired by the design gallery paradigm [23] and sequential image gallery exploration [21]. In our initial design, the user explores a GAN by sequentially selecting images from a current working image gallery (Figure 3.A) using the sequential plane search optimization method [21] to find a photo-realistic image (Figure 3.B). We picked the 5 × 5 grid layout of the current working image gallery without lack of generality and based on recommendations from [21]. The user could submit the current selected photo-realistic image, undo the last image selection (i.e., return to the previous working image gallery), randomize current working gallery (i.e., start over), or change image category (Figure 3.C).

We then piloted a functional prototype (implemented as a Python Django Web application) of our initial design with 26 participants recruited through mailing lists and word of mouth at our academic institution. We used the BigGAN model [6] in our initial prototype implementation, which can generate photo-realistic images from 1,000 different categories. For more details about our choice of the BigGAN model [6] as a test-bed and our user study infrastructure, see Section 4.1. We asked participants to select only photo-realistic images, and followed-up with a subset of them *via* a semi-structured interview in an online chat to collect qualitative data on usability of our initial interface.

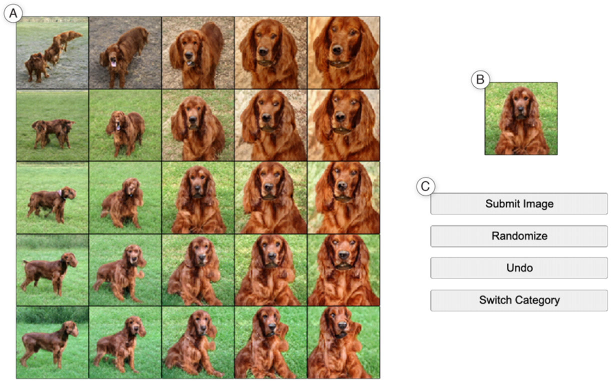


Figure 3: Our *initial* interactive interface for GAN exploration: a) current working image gallery, b) current user-selected image, and c) study navigation tools.

Participants in our formative usability study submitted 528 photo-realistic images (or on average approximately 20 images per participant). They did this by performing sequential image gallery selection 750 times, performed undo 28 times, and randomized their current working image gallery 286 times. Our qualitative analysis of participant responses uncovered two main usability issues: 1) lack of clear mapping between sequential image gallery search method [21] and exploration of a GAN, and 2) lack of user control and freedom. Participants reported that they could not tell what selecting an image from the current working gallery would actually do, and suggested that a conceptual mapping to zoom feature could help. Although participants could undo their last action and even start over by randomizing their current gallery, participants complained that they often lost track of where they were and had no choice but to start over, which caused them annoyance.

We then used insights from our formative usability study and usability principles to carefully iterate on our final interface and interactions design. Figure 2 shows our final interface design and its four main components: a) tool palette, b) current working image gallery, c) gallery snapshots, and d) user-selected quality images. We improved on the sequential image gallery search by mapping it to the zoom in and out paradigm and expanding the set of interactions to include zoom in, zoom out, zoom into region, and pan tools. We improved the user freedom and control *via* landmarks, to allow users to backtrack, in form of gallery snapshots and better visibility of selected images. We describe each in detail below.

3.1.2 Current Working Gallery and Selecting Quality Images.

Our final interface displays a working image gallery with 25 GAN-generated images organized in a 5 × 5 grid (Figure 2.B), at all times. This is the current working gallery that the user can select high-quality images from. To select an image, the user clicks on the checkbox in the upper left corner of the image. The selected image will then appear in the Selected Images region (Figure 2.D) at the bottom of the interface. To remove an image from the list of selected images, the user can click on the checkbox once again, either on the image in the image gallery or in the Selected Images region.

Similarly to [21], we mathematically formalize the current working gallery as a square region of a 2D plane $\mathcal{P}$ (Figure 4). We uniquely define plane region $\mathcal{P}$ in a hyperspace with three vectors $\mathbf{c}$, $\mathbf{u}$, $\mathbf{v}$, where $\mathbf{c}$ is the center of the region of the plane, and $\mathbf{u}$ and $\mathbf{v}$ are two orthogonal vectors with equal length on the plane that both point from the center $\mathbf{c}$. Note that the number of dimensions of each vector $\mathbf{c}$, $\mathbf{u}$, $\mathbf{v}$ corresponds to the number of dimensions of the vector of latent variables $\mathbf{z}$ that the GAN takes as input. We then represent the images in the current working gallery as 25 equally-spaced data points on the plane region, denoted by a set of vectors of latent variables $Z = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_{25}}\}$, where each vector of latent variables $\mathbf{z_i} \in Z$ is an input to the GAN model with the corresponding image as output.

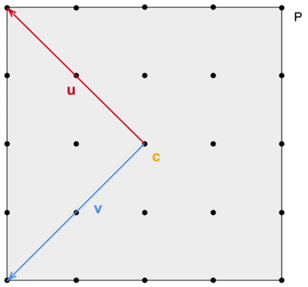


Figure 4: Twenty five equally-spaced data points (denoted by a set of vectors of latent variables $Z = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_{25}}\}$) from a square plane region $\mathcal{P}$ defined by three vectors $\mathbf{c}, \mathbf{u}, \mathbf{v}$. Note that $\mathbf{c}$ denotes the central image in the plane region.

To start, the interface generates an initial current working gallery by sampling vectors $\mathbf{c}$, $\mathbf{u}$, and $\mathbf{v}$ from a normal distribution, where we compute the projection of vectors $\mathbf{u}$ and $\mathbf{v}$ one onto the other and then resize them to a fixed length. This results in a random orientation of the plane. In our example, we empirically estimated these initial parameters to be $\mathbf{c} \sim \mathcal{N}(0, 0.1)$, $\mathbf{u} \sim \mathcal{N}(0, 1)$, $\mathbf{v} \sim \mathcal{N}(0, 1)$, and we fix the length of vectors $\mathbf{u}$ and $\mathbf{v}$ to 15. We then pick 25 equally spaced points on the plane segment corresponding to 25 latent vectors $\mathbf{z_i} \in Z$ (i.e., 25 different GAN-generated images in the current working gallery).

Note that the current working gallery does *not* show only quality images. Instead, it allows the user to explore the space of possible GAN-generated images (both with high and poor quality), using eight different tools in the tool palette (Figure 2.A): zoom in, zoom out, zoom into region, pivot, snapshot, randomize, and undo and redo. Additionally, the user can pan the images in the current working gallery using arrows on the sides of the current working gallery. We describe each tool in detail below.

3.1.3 Zoom In and Zoom Out.

To explore the search space around a particular image of interest in the current working gallery, the user can click on the *Zoom in* or *Zoom out* tools in the tool palette and then click on the image. The resulting working gallery will have the image that the user clicked on in the center and it will be surrounded by images that are more similar to it than the images in the current working gallery (Figures 5 and 6). The only difference between *Zoom in* and *Zoom out* is that, in the case of *Zoom in*

(Figure 5), the images in the four corners of the resulting working gallery will be more similar to the center image than the four corner images in resulting gallery after *Zoom out* (Figure 6). Note, however, that *Zoom out* tool is not a simple "undo" of the *Zoom in* tool because the user could use other tools after zooming into an image, or zoom out of an image that the user has not previously zoomed into.

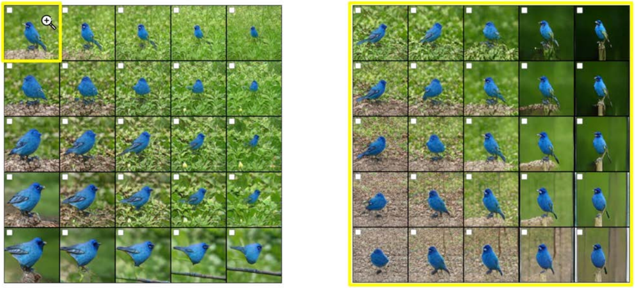


Figure 5: An example of zooming-in on the top-left image in a current working gallery (left). The zoomed-in image is in the center of the resulting working gallery (right) and is surrounded by more similar images than before.

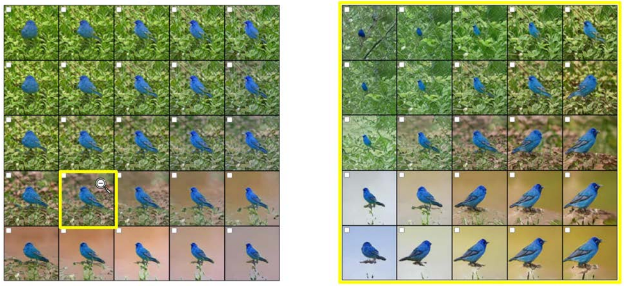


Figure 6: An example of zooming-out of the image at the fourth row, second column in a current working gallery (left). The zoomed-out image is in the center of the resulting working gallery (right) and is surrounded by more diverse images than before.

We implement the *Zoom in* and *Zoom out* tools based on an existing method [21], which enables the user to sequentially enhance and select a single highest quality image from a sequence of image galleries using Bayesian optimization [5]. Our insight is that we can adopt their original method to find a set of similar GAN-generated images that the user has current interest in exploring, even if the image the user zooms into or out off is not the highest quality image in the current working gallery.

Let the plane segment $\mathcal{P}$ correspond to the current working gallery, and suppose the user clicks on an image (with a corresponding vector of latent variables $\mathbf{z}^* \in Z$) in the current gallery with one of the *Zoom in* or *Zoom out* tools. Our goal is to construct the resulting working gallery and its corresponding plane segment $\mathcal{P}'$, which is determined by vectors $\mathbf{c}', \mathbf{u}', \mathbf{v}'$, such that the resulting plane segment $\mathcal{P}'$ contains images that are more "similar" to the image the user clicked on than the images on the current working gallery plane segment $\mathcal{P}$.

To do so, on each click, we run a single iteration of Koyama et al.'s [21] Bayesian optimization method on the vector of latent variables $\mathbf{z}^*$ to compute vectors $\mathbf{c}', \mathbf{u}', \mathbf{v}'$. This results in the image that the user clicked on as the central image in plane segment $\mathcal{P}'$ (i.e., $\mathbf{c}' = \mathbf{z}^*$), and vectors $\mathbf{u}'$ and $\mathbf{v}'$ such that they increase the expected improvement in similarity between the central image $\mathbf{z}^*$ and the rest of the images on the new plane segment $\mathcal{P}'$ compared to the original plane segment $\mathcal{P}$. After we compute vectors $\mathbf{c}', \mathbf{u}', \mathbf{v}'$, we scale the area of the resulting plane region (i.e., the length of vectors $\mathbf{u}'$ and $\mathbf{v}'$) by an empirically determined scaling factor k. In case of *Zoom in* we shorten the vectors so the area decreases; and in case of *Zoom out* we lengthen the vectors, so the area increases.

Note that unlike in [21], we restart the iterations for subsequent invocations of *Zoom in* or *Zoom out* tools, since the user interest in a particular group of images can change. This in turn keeps our computational complexity low and the diversity of images in the resulting working galleries high. For specific details on how to implement Koyama et al.'s [21] Bayesian optimization method, and any related proofs, please see [12, 21, 22].

3.1.4 Zoom Into Region.

Here, we describe the *Zoom into region* tool separately from the *Zoom in* and *Zoom out* tools because of their fundamental differences. When the user clicks on an image in the current working gallery with the *Zoom into region* tool and drags the cursor to another image it forms a region. When the user clicks with the tool on the other image, the top-left image and bottom-right image in the region will become the top-left and bottom-right images in the resulting working gallery (Figure 7). If the user clicks on the same image with the tool twice, it will result in no changes in the current working gallery.

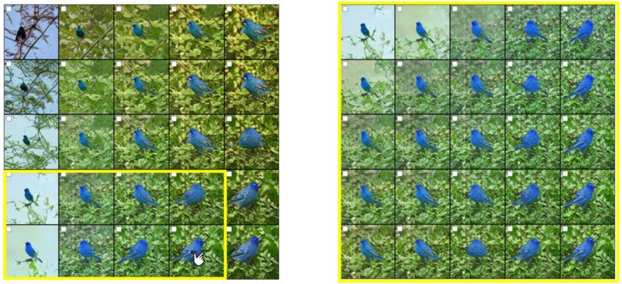


Figure 7: An example of zooming-into-region of the bottom-left eight images in a current working gallery (left). The top-left and the bottom-right images in the resulting working gallery (right) correspond to the images in top-left and bottom-right corner of the region in the current working gallery. Note that the resulting working gallery is a square despite the zoom in region being a rectangle.

Mathematically, let the current working gallery be defined by a plane region $\mathcal{P}$, and let the vectors of latent variables corresponding to the top-left and bottom-right images in the zoom in region be $\mathbf{z_i}$ and $\mathbf{z_j}$, respectively. To construct the resulting working gallery plane region $\mathcal{P}'$ by computing vectors $\mathbf{c}', \mathbf{u}', \mathbf{v}'$ using the following equations:

$$\mathbf{c}' = \frac{\mathbf{z_i} + \mathbf{z_j}}{2} \tag{1}$$

$$\mathbf{u}' = \frac{\mathbf{z_i} - \mathbf{z_j}}{2} \tag{2}$$

$$\mathbf{v}' = \frac{||\mathbf{u}'||}{||\mathbf{v_1} - \mathbf{v_2}||}(\mathbf{v_1} - \mathbf{v_2}) \tag{3}$$

where

$$\mathbf{v_1} = \mathbf{z_i} - 2\mathbf{z_j}$$

$$\mathbf{v_2} = \frac{\mathbf{v_1} \cdot \mathbf{u}'}{\mathbf{u}' \cdot \mathbf{u}'}\mathbf{u}'$$

In Equation (3), $\mathbf{v_1} - \mathbf{v_2}$ is the vector rejection of $\mathbf{v_1}$ from $\mathbf{u}'$, where $\mathbf{v_1} - \mathbf{v_2}$ ensures that $\mathbf{v}'$ is perpendicular to $\mathbf{u}'$, while the $\frac{||\mathbf{u}'||}{||\mathbf{v_1} - \mathbf{v_2}||}$ term scales the vector so that $||\mathbf{u}'|| = ||\mathbf{v}'||$. Our choice of $\mathbf{v_1}$ ensures that $\mathbf{v_1} \nparallel \mathbf{u}'$ (otherwise, $\mathbf{v}'$ would become zero), while $\mathbf{v_2}$ computes the vector projection of $\mathbf{v_1}$ from $\mathbf{u}'$. Notice that the resulting plane region is always square, even if the zoom-in-region in the interface is rectangular (Figure 8). Also notice that, unlike *Zoom in* and *Zoom out* tools, *Zoom into region* tool does not change the orientation of the resulting plane region.

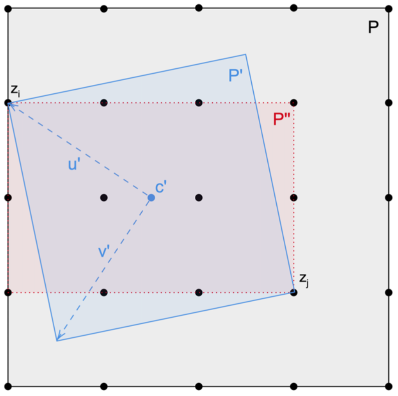


Figure 8: A square plane region $\mathcal{P}$ corresponding to the current working gallery, zoom-in region $\mathcal{P}''$, and the resulting plane region $\mathcal{P}'$ defined by three vectors $\mathbf{c}', \mathbf{u}', \mathbf{v}'$. Note that the resulting region preserves square shape, despite rectangular shape of the zoom-in region $\mathcal{P}''$.

3.1.5 Pivot. Up until now, all tools focused on narrowing down to a similar set of images. However, to increase diversity of images, sometimes it may be beneficial to break away from similar images to search in underexplored areas of image space. However, simply restarting the exploration from a random working gallery is not always desirable as it may prevent explorations near the current search space region.

Thus, we added the *Pivot* tool to the tool palette. When the user clicks on the *Pivot* tool, the resulting working gallery shows a different set of images centered at the same image as the current working gallery (Figure 9). Note that despite its user-friendly name that describes the appearance of the effect in the resulting gallery, mathematically, this tool actually shifts the resulting plane region.

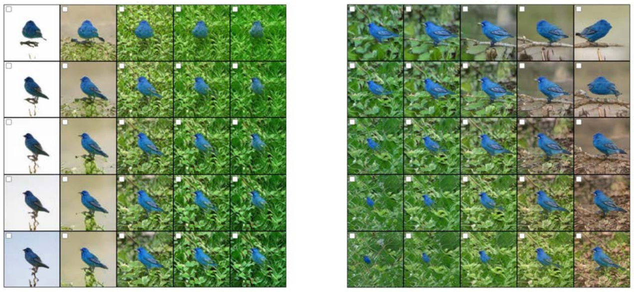


Figure 9: An example of pivoting in a current working gallery (left). Note that the resulting working gallery (right) is still centered at the same image, and the rest of the images "pivoted" around it.

As mentioned earlier, there are 25 images in the current working gallery (each corresponding to a point on the $Z = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_{25}}\}$ in the current plane region $\mathcal{P}$). Each $\mathbf{z_i} \in Z$ has a corresponding vector of latent variables with 120 dimensions. Thus, fixing the same subset of dimensions in all 25 latent vectors and changing the values of the remaining dimensions results in new, shifted plane region $\mathcal{P}'$.

3.1.6 Pan. There are four arrows surrounding the image gallery. The user can click on them to pan or move the current visible area of the gallery in the direction of the arrow (Figure 10). Note that the current working gallery only shows a region of the current plane 2D, but the plane extends in all four directions. When the user clicks on one of the four arrows surrounding the current working gallery, the region of the plane moves by one row or column of images (depending on the direction of the pan).

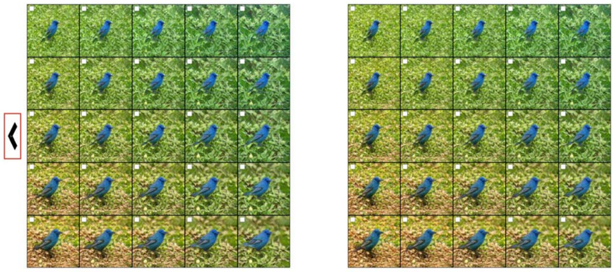


Figure 10: An example of panning the current working gallery (left) to the left, with the resulting working gallery (right) showing that the region of the working gallery moved by one column of images to the left.

Here we explain our formalization of the *Pan* command on the example of panning to the left (without loss of generality). Let the images in the current working gallery as a set of vectors of latent variables $Z = \{\mathbf{z_1}, \mathbf{z_2}, ..., \mathbf{z_{25}}\}$ on the current plane region $\mathcal{P}$ defined by vectors $\mathbf{c}, \mathbf{u}, \mathbf{v}$, and the resulting gallery after panning, with the plane region $\mathcal{P}'(\mathbf{c}', \mathbf{u}', \mathbf{v}')$. Then, we can define the following two direction vectors:

$$\mathbf{a} = \mathbf{z}_6 - \mathbf{z}_1 \tag{4}$$

$$\mathbf{b} = \mathbf{z}_2 - \mathbf{z}_1 \tag{5}$$

Intuitively, vector $\mathbf{a}$ points from the top of the gallery to the bottom of the gallery, and vector $\mathbf{b}$ points from gallery's left to gallery's right. Now, we can compute for $\mathbf{c}', \mathbf{u}', \mathbf{v}'$ as:

$$\mathbf{c}' = \mathbf{z}_{12} \tag{6}$$

$$\mathbf{u}' = 2(-\mathbf{a} - \mathbf{b}) \tag{7}$$

$$\mathbf{v}' = 2(\mathbf{a} - \mathbf{b}) \tag{8}$$

Similarly, it follows that we can panning in the other three directions using the same way to construct the resulting plane region of the resulting working gallery.

3.1.7 User Control and Freedom: Gallery Snapshot, Randomize, Undo and Redo. To support user control and freedom [24], we provide tools to save an exploration point to restart at a later time or to start over. Clicking on the *Snapshot* tool records the current working gallery and allows the user to return to it at any point in the exploration. The *Randomize* tool will simply generate a new working gallery (Figure 11) using the same method we used to initialize the first working gallery in Section 3.1.2. Finally, we provide the standard *Undo* tool, which will change the current working gallery into its previous state; and *Redo*, which will repeat any commands that the user undid in the previous step.

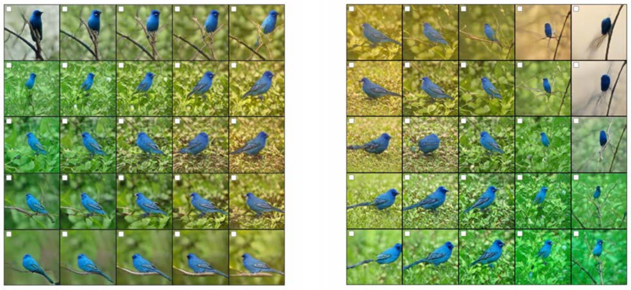


Figure 11: Performing randomize operation to the gallery.

3.2 Automatically Generated Image Galleries

The existing methods to generate diverse set of photo-realistic images from GANs sample the latent variable $\mathbf{z}$ from a truncated normal distribution, which is different from using a normal distribution during model training phase. However, not all GANs are amenable to such sampling using truncation, which also requires visual examination of images to find a reasonable, but still subjective threshold of the truncated normal distribution for each model.

Our method aims to solve those challenges by sampling from the posterior probability distribution of image parameters given a sample of quality images and their corresponding latent variables selected using our interactive interface. To sample images from the posterior distribution, we use MCMC [2] as one of few computationally feasible, well-established, principled sampling methods.

3.2.1 Markov Chain Monte Carlo (MCMC). It is difficult to guarantee the quality and diversity of images generated from a narrow prior probability distribution. Instead, having collected samples of quality images using our interface, we can estimate a posterior probability distribution of the latent variable $\mathbf{z}$ in a principled way using a MCMC method [2].

We model the probability distribution of the latent variable $\mathbf{z}$ in a mixture of two Gaussians: one component is to capture the photo realistic aspect, and the other component is to support diversity of images. We assign the following prior values for the model parameters: the prior weights of two mixtures follow a Dirichlet distribution with concentration parameters $(1, 1)$; the prior means of two mixtures both follow a normal distribution $\mathcal{N}(0, 1)$; the prior standard deviations of two mixtures both follow an inverse gamma distribution $\mathcal{IG}(1, 1)$. We formulated this using the Bayes' theorem:

$$P(\theta|Z) = \frac{P(Z|\theta)P(\theta)}{P(Z)} \tag{9}$$

where $Z = \{\mathbf{z_i}\}_{i=1}^n$ is a set of n observations, and θ are model parameters. $P(\theta)$ is the prior probability distribution of the parameters θ, $P(Z|\theta)$ is the likelihood of the observed data Z, and $P(\theta|Z)$ is the posterior distribution of θ given observations Z.

We have declared $P(\theta)$, and we also have the close-form formula for $P(Z|\theta)$ since we have chosen the probability distribution to model $\mathbf{z}$. Now the difficulty of deriving the posterior distribution $P(\theta|Z)$ is to compute $P(Z)$:

$$P(Z) = \int_{\Theta} P(Z|\theta)P(\theta)d\theta = \int_{\Theta} P(Z,\theta)d\theta \tag{10}$$

It is difficult to derive a close-form solution for Equation (10), so we use MCMC to estimate the posterior distribution $P(\theta|Z)$. MCMC begins by randomly picking a parameter setting. The simulation then samples more random points according to an algorithm (*e.g.* Metropolis-Hastings, Gibbs, and NUTs) and adds them to the sequence of parameter values with a given probability, to ensure it visits the high posterior probability regions more often. This way, the number of sampling points in each region is proportional to $P(\theta|Z)$. The final histogram of the sampled points gives a good estimation of the posterior distribution.

Suppose θ_0 is the current parameters, and θ is the next proposed parameters. If we take the ratio of the posterior of θ to the posterior of θ_0, $P(Z)$ in the Bayes' theorem can be canceled out:

$$\alpha = \frac{P(\theta|Z)}{P(\theta_0|Z)} = \frac{\frac{P(Z|\theta)P(\theta)}{P(Z)}}{\frac{P(Z|\theta_0)P(\theta_0)}{P(Z)}} = \frac{P(Z|\theta)P(\theta)}{P(Z|\theta_0)P(\theta_0)} \tag{11}$$

In other words, we can compare the posterior of two parameter settings relatively. With a Metropolis-Hastings sampler, α is referred to as the acceptance rate. A uniform random number $u \in [0, 1]$ will then be generated and compared with α. If $u \leq \alpha$, the proposed parameters θ will be accepted. If $u > \alpha$, the proposed θ will be rejected and the current parameters θ_0 kept for the next iteration.

4 EXPERIMENTS

Here, we describe three experiments (each a separate crowdsourced user study) that we conducted to evaluate our proposed method for exploring and sampling quality images from a GAN. We implemented a functional prototype of our interface for the purpose of the user studies. In each of the three experiments, we used the BigGAN model [6] as a test bed. Thus, in our user studies, image quality refers to photo-realism since the goal of BigGAN model [6] is to generate diverse photo-realistic images. We conducted all three user studies on the Amazon Mechanical Turk (MTurk) [1] and compensated the participants proportional to the minimum wage at our location. Our user studies were deemed exempt (HUM00184499) by our Institutional Review Board (IRB).

Our ultimate goal was to compare our automatically generated galleries against the current, state-of-the-art qualitative GAN evaluation method [6] in our last user study. Although we built up to this comparison through our three studies, we did not iterate on our interface design between the three studies. In our first study we investigated participant interaction with our interface and its ability to support them in exploring and selecting a diverse set of photo-realistic images from the BigGAN model [6]. In the second study, we validated the output of our first study to account for limitations of crowdsourced studies. In the third study, we compared against a baseline from BigGAN evaluation [6], which used only static randomly sampled image galleries to visually examine output from a GAN.

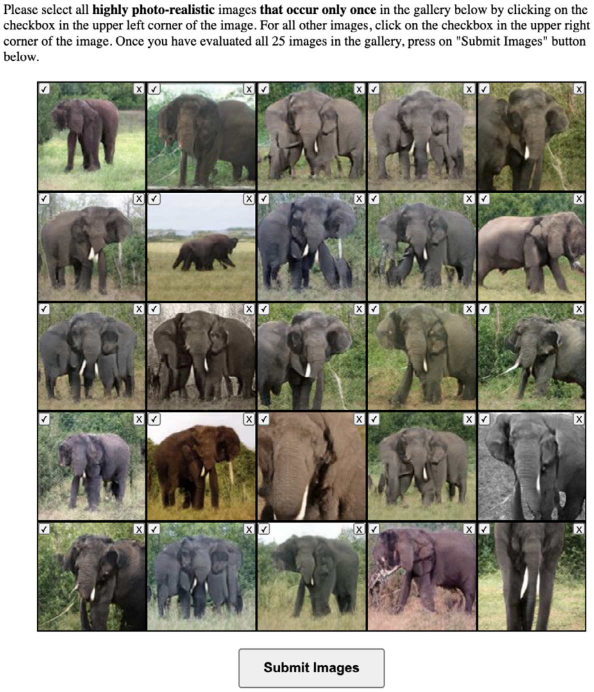
4.1 User Study Software and Implementation

To be able to run our user studies, we required a highly functional implementation of our proposed method. Here, we describe different components we have implemented.

4.1.1 Generative Adversarial Network (GAN) Model. To illustrate our method, we used the BigGAN model [6] in our prototype implementation, which can generate photo-realistic images from 1,000 different categories. Thus, in this case, we judge quality of images based on how photo-realistic they are. We used the publicly available Python implementation of the BigGAN model [7] trained on ImageNet [11], a large scale image dataset, at $128px \times 128px$ resolutions, in which the vector of latent variable $\mathbf{z}$ has 120 dimensions. We specifically selected this model because it attempts to push the boundaries of capabilities of GANs. This means that the model does not necessarily always generate photo-realistic images for all categories, making it a perfect test bed for our experiments. Also, it would be difficult to explore this large high-dimensional sample space manually even for a single category.

We selected ten different BigGAN categories (*Cardoon*, *Cup*, *Dipper*, *Dome*, *Headland*, *Indigo finch*, *Irish Setter*, *Monarch Butterfly*, *Poncho*, and *Tusker*). We selected these categories based on two criteria: 1) they were used as exemplars to visually examine and illustrate the capabilities of the BigGAN model [6] (e.g., *Irish setter*, *Monarch Butterfly*), or 2) they showed exceptionally poor quality in our initial visual examination of the model (e.g., *Cup*, *Tusker*).

4.1.2 Study Prototype Implementation. We implemented a fully functional interactive prototype of our design (Figure 2) as a Python Django Web application and used it in our first user study. The public facing Web application communicated with a MySQL database server, where we stored all our application and user data, over a secure connection.



Please select all **highly photo-realistic** images **that occur only once** in the gallery below by clicking on the checkbox in the upper left corner of the image. For all other images, click on the checkbox in the upper right corner of the image. Once you have evaluated all 25 images in the gallery, press on "Submit Images" button below.

Figure 12: Interface of our image gallery labeling tool. The user can click on a top-left checkbox to indicate that an image is photo-realistic, or on the top-right checkbox to indicate the image is not photo-realistic.

Note that Bayesian inference is computationally expensive on very high dimensional spaces [21], such as the one we intended to explore. Such poor performance could impact the interactivity of our prototype. Therefore, we set the number of dimensions of the vector that the tools in our interface can operate on at any given time to 10. Although we randomly pick these 10 dimensions at our prototype start time, the user can use the *Pivot* tool to pick another 10 dimensions the other tools will operate on.

4.1.3 Image Sampling Implementation. In our study, we implemented the MCMC sampling in Python with PyMC3 [28]. We used a mixture of two Gaussians: one to capture photo-realistic aspects of sampled images, and the other to model diversity in sampled image. We set the prior weights of two mixtures to follow a Dirichlet distribution with concentration parameters $(1, 1)$; the prior means of two mixtures both follow a normal distribution $\mathcal{N}(0, 1)$; the prior standard deviations of two mixtures both follow an inverse gamma distribution $\mathcal{IG}(1, 1)$. We chose these priors to match the training parameters of the BigGAN model [6]. In our implementation, we use the No-U-Turn sampler (NUTs) [16], which is the default step method used by PyMC3 [28]. When sampling each dimension of $\mathbf{z}$, we tune for 2,000 samples and then draw 4,000 posterior samples.

4.1.4 Image Labeling Software. In addition to our interactive interface, we also implemented a separate image gallery labeling tool (Figure 12), which we used in the other two user studies. The labeling tool was similar to the working gallery portion of our interface (Figure 2.B), but without any of the tools. Instead, the user could only select a checkbox to indicate that an image was photo-realistic, or select another checkbox to indicate the image was not photo-realistic. To quickly evaluate the galleries and to minimize any delays, we pre-generated the galleries and stored the sampled images in the database for quick retrieval. We then randomly assigned the galleries to participants in our second two studies based on the study protocol.

4.2 Interactive Model Exploration User Study

In this study, we evaluated the ability of our interface to support exploration and selection of a diverse set of photo-realistic images from the BigGAN model [6].

4.2.1 Task and Method. Participants joined our study by clicking on an MTurk Human Intelligence Task (HIT) from a list of available HITs. The participants then had to view the description of our task, and read and accept our study consent form. After consenting, participants had to view an instruction video that described our interface and illustrated how to use each of the tools, before proceeding to the study task.

In the study task, we instructed the participants to explore the working galleries for a specific image category using all available tools, and select 10 photo-realistic images. We randomly assigned participants to image categories. Participants could only submit their HIT once they selected at least 10 images, but the interface did not prevent them from selecting more images. We limited each HIT to 20 minutes. Participants could come back and use the tool to find images in other categories (i.e., complete more than one HIT over multiple sessions).

In this study, we counted the number of times participants interacted with different tools to investigate their interaction with the tool and show what tools contributed to finding photo-realistic images. We also counted the number of images participants selected across galleries to show the effectiveness of our interface. We did not discard any images that the participants selected. We differed judgement about photo-realism of the selected images until our next user study. Note that we did not ask participants to use an existing state-of-the-art, non-interactive, randomly generated image galleries where they have *no* control over what images they are looking at (such as in [6]), since such comparison is not necessary to show that having an interactive tool (compared to no interactive tool) is useful.

4.2.2 Participants. We recruited 367 participants. All participants were located in the United States, and were ages 18 and above. Only participants who had more than 100 HITs approved on MTurk (with an approval rate greater than 95%) could take part in this study. We compensated each participant $2.50 per HIT, which resulted on average in $10/h.

4.2.3 Results. Over 956 individual sessions, participants panned current working galleries 2,138 times, zoomed in, zoomed out, and zoomed into region of working galleries 969, 107, and 210 times respectively. They pivoted their current working galleries 150 times and started from a new random working gallery (i.e., randomized) 679 times. They took 200 snapshots and reverted to a previous snapshot 34 times, and used undo and redo 97 and 15 times respectively. This suggests that the tools enabled participants to select images beyond those present in the starting randomly generated working gallery they first looked at (i.e., equivalent to random image gallery exploration).

Table 1: Number of collected images per Category

Category	Number of Images
Cardoon	867
Cup	888
Dome	908
Dipper	1,018
Headland	962
Monarch Butterfly	1,069
Indigo Finch	1,061
Irish Setter	981
Poncho	1,067
Tusker	1,205
Total:	**10,026**

Participants selected 10,026 GAN-generated images using our prototype. The number of collected images in each category ranged from 867 for *Cardoon* to 1,205 for *Tusker* (Table 1). Our results show that each participant was on average able to find about 27 photo-realistic images. This, together with fairly balanced number of images across categories, suggests that the participants were able to successfully use our interface to explore and select images they thought were photo-realistic, the primary way to judge the quality of images generated using the BigGAN model [6].

4.3 GAN-generated Image Quality Evaluation

In this study, we evaluated photo-realism of GAN-generated images that participants selected in our first study. Due to large number of images, it is impractical to begin visually examination of the GAN model at this stage. Instead, in this study we first crowdsource labels to identify the best photo-realistic images that we collected in the first study.

4.3.1 Task and Method. Participants joined our study by clicking on an MTurk Human Intelligence Task (HIT) from a list of available HITs. The participants then had to view the description of our task, and read and accept our study consent form. After consenting, participants proceeded to the study task.

In the study task, we instructed the participants to select only photo-realistic images from an image gallery using our image labeling tool. Each image gallery contained 25 images from the same image category. For each task, we randomly selected a category and then randomly selected 25 images from that category from 10,026 images participants generated using our prototype in the first study. We limited each HIT to 5 minutes. Participants were *not* allowed to complete more than one HIT in this study.

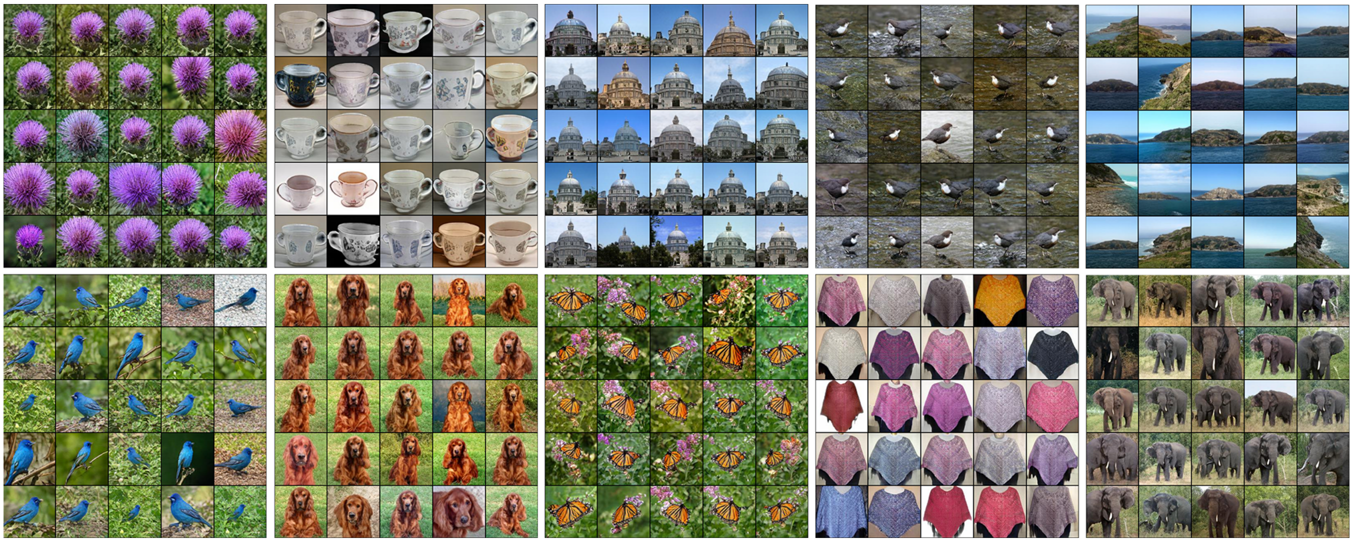


Figure 13: Ten different randomly selected image galleries containing top photo-realistic GAN-generated images that participants selected using our interactive prototype in the first study.

Table 2: Ratings of Photo-realistic Images per Category

Category	Number of Raters	Krippendorff's alpha	Images with 1 or more Photo-realistic Label
Cardoon	139	0.0677	82.93%
Cup	147	0.0917	74.44%
Dome	170	0.066	87.22%
Dipper	178	0.0997	83.40%
Headland	174	0.0588	91.06%
Monarch Butterfly	175	0.0426	84.75%
Indigo Finch	165	0.164	76.44%
Irish Setter	162	0.101	77.37%
Poncho	179	0.104	86.41%
Tusker	133	0.0754	59.75%

We then counted the number of images that participants labeled as photo-realistic and calculated their agreement using the Krippendorff's Alpha Reliability Coefficient to measure the quality of images across galleries. We did not discard any images that the participants labeled. Instead, we used our labeling tool interface to visually examine and qualitatively assess images that the participants marked as photo-realistic.

4.3.2 Participants. We recruited 1,622 participants in this study. All participants were located in the United States, and were ages 18 and above. Only participants who had more than 100 HITs approved on MTurk (with an approval rate greater than 95%) could take part in this study. We compensated each participant $0.25 per HIT, which resulted on average in $7.50 per hour.

4.3.3 Results. Each of 10,026 images generated using our tool in the first study had at least 2 labels. Out of the 10,026, 8,015 images (79.94%) were rated as photo-realistic by at least one participant. The percentage of images rated as photo-realistic by at least one participant in each category ranged from 59.75% for *Tusker* to 91.06% for *Headland* (Table 2). Though the Krippendorff's Alpha in Table 2 indicates a lot of noise in agreement (e.g., the ratings could be highly subjective, there could be presence of too lenient or too harsh raters), we interpret the high percentage of images participants labeled as photo-realistic to suggest that our prototype could support exploration and selection of quality images, despite the limitations of the BigGAN model.

We then performed visual examination of images that more than 75% of participants labeled as photo-realistic. Figure 13 shows a selection of GAN-generated images that participants selected using our interactive prototype in the first user study. We found that the participants were able to identify a diverse set of photo-realistic images that go beyond the homogeneous set of images used in the original visual examination [6] of the BigGAN model. However, we also identified what appears to be a systemic problem with the BigGAN model: it fails to generate photo-realistic images for categories where objects in images are asymmetric (e.g., *Cup*, *Tuskar*). We further investigate this in our next study.

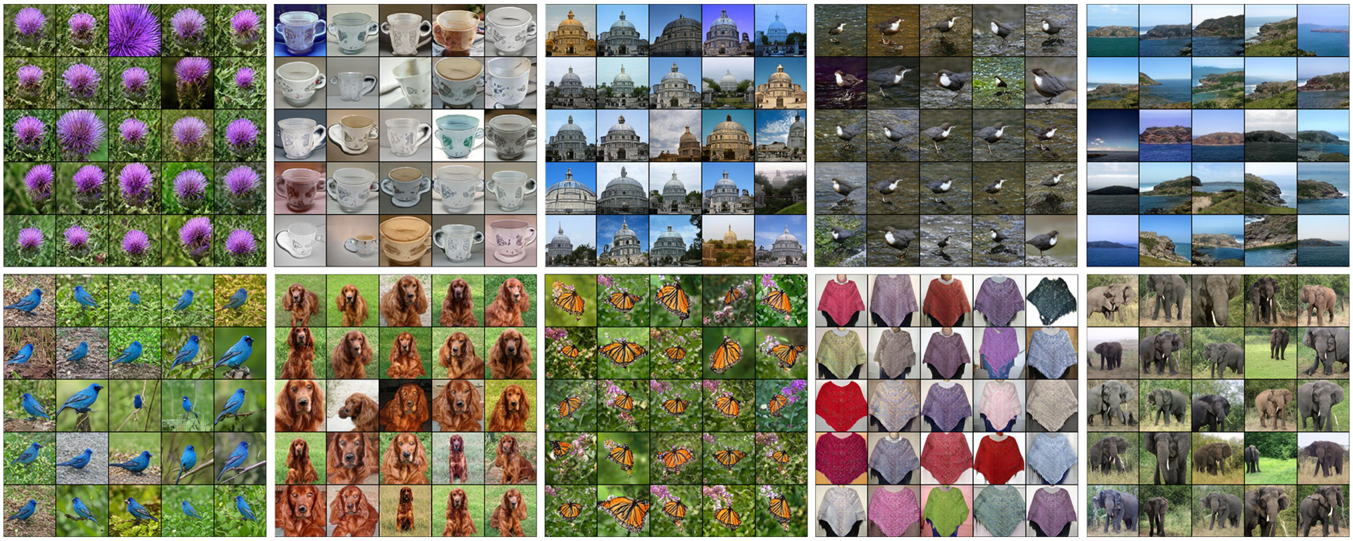


Figure 14: Ten different image galleries containing images sampled from the BigGAN model using our sampling method.

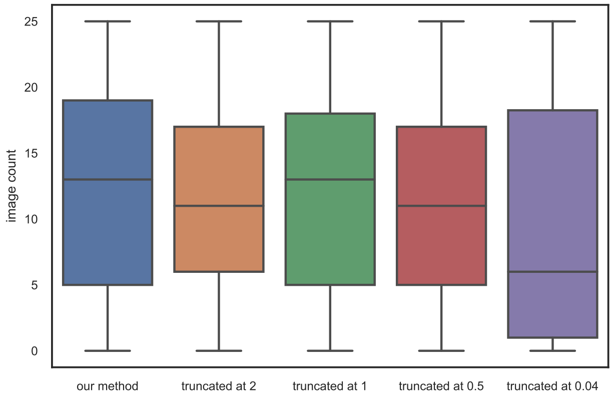


Figure 15: The number of unique photo-realistic images in galleries sampled using our method and the 4 baselines.

4.4 Automatically Sampled Images Validation

Although our first two studies showed that our prototype can support exploration and selection of photo-realistic images from the BigGAN model [6], such approach is not scalable, as the number of photo-realistic images we can collect in this way is bounded by the number of participants in our studies and the time it takes them to select images. Thus, we evaluated the ability of our method to sample photo-realistic images from the BigGAN model and compared it with the existing baseline sampling methods from [6].

4.4.1 Task and Method. Participants joined our study by clicking on an MTurk Human Intelligence Task (HIT) from a list of available HITs. The participants then had to view the description of our task, and read and accept our study consent form. After consenting, participants proceeded to the study task.

In the study task, we randomly assigned participants into one of 50 conditions where each condition was a combination of (*Method* × *Category*). *Method* included our sampling method, and 4 baselines. In our sampling method (Section 4.1.3), we used parameters of GAN-generated images from our first study that had 75% or more participants in our second study agree that they are photo-realistic. Each baseline sampled a vector of latent variables $\mathbf{z}$ from a truncated normal distribution with mean 0 and standard deviation 1, but with four different truncation levels as suggested in [6]: $(2, 1, 0.5, 0.04)$. *Category* included the ten BigGAN image categories.

We then instructed the participants to select photo-realistic images that occur only once in their assigned image gallery using our image labeling tool, and to deselect all others. We asked them this to evaluate if a *Method* can produce a variety of photo-realistic images. We limited each HIT to 5 minutes. Participants were *not* allowed to complete more than one HIT in this study.

We then counted the number of images that participants labeled as non-repeating, photo-realistic images to measure the quality of sampled galleries. We ran the Align Rank Transform (ART) [32] to analyze the data. We used this statistical method because we had more than one independent variable (*Method* × *Category*) and our ordinal data was not normally distributed. We performed a pairwise contract test as is customary with ART [32] to find the differences between conditions in case of a main effect or interactions. Our goal was to show the effect of *Method* on the number of diverse photo-realistic images it produces, and not necessarily to show *Method* × *Category* interaction. Instead, our study design accounted for the effect of different categories, and we reported it for transparency.

4.4.2 Participants. We recruited different 1,000 participants in this study. All participants were located in the United States, and were ages 18 and above. Only participants who had more than 250 HITs approved on MTurk (with an approval rate greater than 95%) could take part in this study. We compensated each participant $0.25 per HIT, which resulted on average in $7.50 per hour.

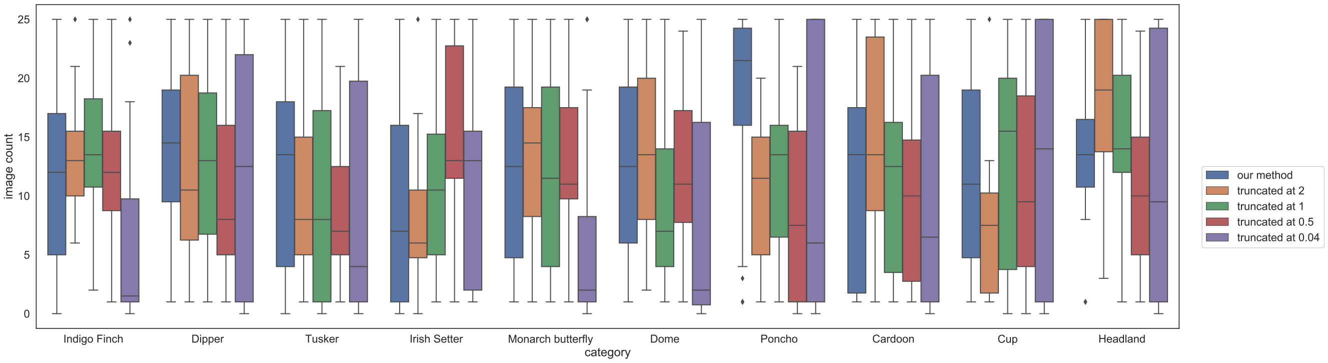


Figure 16: The number of unique photo-realistic images in galleries from 10 different categories sampled using our method and the 4 baselines.

4.4.3 *Results.* The results of our statistical tests found significant main effect of *Method* on the number of selected non-similar photo-realistic images ($p = 0.0006$) (Figure 16), but could not find a significant effect of *Category* ($p = 0.08$); the test also found significant interaction between the two ($p = 0.0138$). Our contrast test found a pairwise difference only between our method and truncation 0.04 ($p = 0.0002$), truncation 2 and truncation 0.04 ($p = 0.0357$), and truncation 1 and truncation 0.04 ($p = 0.0191$). Not surprisingly, our contrast test could not find pairwise interactions (Figure 15), likely due to large number of pairwise comparisons compared to number of participants in each condition. This shows that our principled sampling method outperforms at least one of the arbitrary baseline thresholds from [6].

We then performed visual examination of image galleries generated using our method and the four baselines. Figures 17 and 18 show example comparison of randomly selected image galleries generated using our method and the four baselines for top-rated and bottom-rated five categories respectively. In our visual examination, we found that our method consistently sampled more diverse galleries of photo-realistic images compared to the baselines. We also found that the photo-realism and diversity of images varied across categories and baselines, which could have contributed to wide distribution of image counts across baseline thresholds (Figure 15). This showed additional evidence in favor of our principled sampling method for each image category compared to arbitrary baseline thresholds.

5 DISCUSSION

We have shown that our method could support exploration and sampling of diverse set of photo-realistic images from a GAN model, even when the GAN model struggles with generating photo-realistic images in majority of cases (such is the case with the BigGAN model and its *Tusker* category). Our validation showed not only that each participant in our first study could explore the model enough to on average select 27 images, but also that those images are likely to be highly photo-realistic for most categories.

The results of our studies showed that our principled image sampling approach is better than a heuristic approach where it is unclear what sampling threshold one should pick. Our results in the third study showed evidence that our method outperformed sampling from narrow probability distributions that favor photo-realism over diversity. This is because a low scoring gallery can have highly photo-realistic images, but all images might be too similar. It also highlighted that the users may favor diverse galleries over highly photo-realistic ones.

However, results from our user studies echo concerns from previous work [27, 34] that suggested that the user feedback (especially from crowdsourced user studies) is often too noisy for accurate GAN evaluation. In our studies, we have also identified participants that consistently rated too many or too few images as photo-realistic, even when their feedback was objectively too lenient or too harsh. Although this strengthens the argument for better quantitative GAN evaluation methods, the need for qualitative visual examination by humans remains indispensable, and better qualitative assessment methods may be required.

Note that our goal was not to show that our method can always sample photo-realistic images, but to show that it can help identify both capabilities and limitations of a GAN model. For example, visual examination of image galleries generated using our method showed that BigGAN is better at some categories than others. This is because BigGAN model is experimental and is pushing the limits of how many categories a model can learn. Some of the categories are inherently bad in BigGAN (especially ones with asymmetric objects). However, participants in our first study were able to locate reasonable examples using our prototype and our sampling method was able to generate photo-realistic image galleries even for those problem categories.

Visual examinations of images generated using our method and comparison with the baselines showed that the existing visual examination methods might have sampled images from too wide or too narrow distributions, which does not necessarily accurately reflect the capabilities of a GAN model. When the distribution is too wide, the sampled images were not photo-realistic; when it was too narrow, they all looked too similar and were not diverse. However, there is no way to tell how to manually pick the best baseline threshold for each image category. Instead, our method automatically estimates the probability distribution from which it samples highly diverse set of photo-realistic images, including from

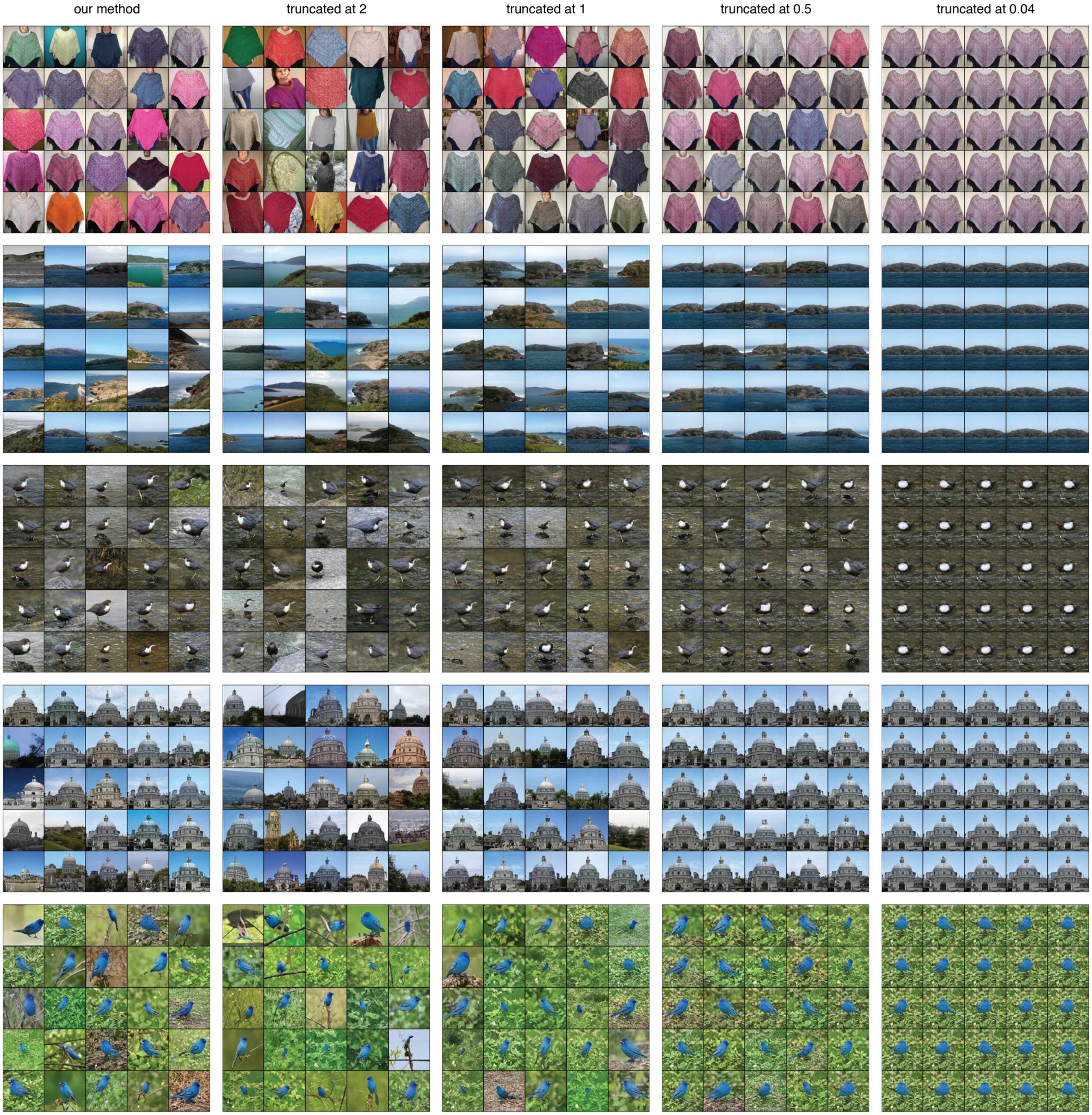


Figure 17: Example randomly selected image galleries generated using our method and the four baselines for five categories with on average the most diverse set of photo-realistic images rated by participants in our third study.

regions of the search space where one would not expect them to be (e.g., far from the mean of a vector of latent variables **z**). Thus, our method provides a better insight into more regions of search space to offer a more comprehensive qualitative evaluation of a GAN.

6 CONCLUSION AND FUTURE WORK

In this paper, we presented an interactive method for exploring and qualitatively validating a GAN. We presented an interface that provides a number of new tools for interactive GAN exploration. Our method offers a comprehensive exploration and qualitative

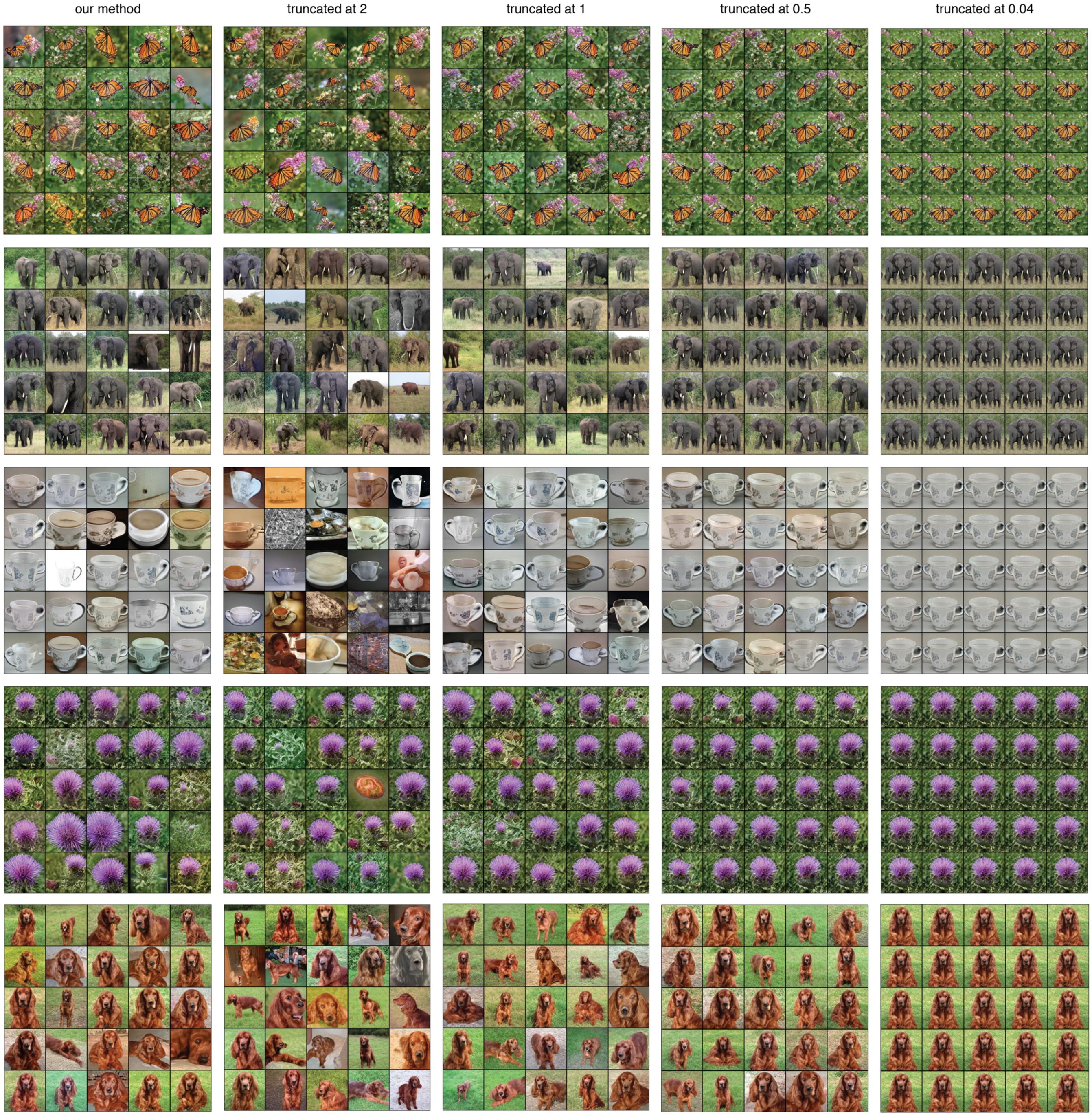


Figure 18: Example randomly selected image galleries generated using our method and the four baselines for five categories with on average the least diverse set of photo-realistic images rated by participants in our third study.

evaluation of a GAN through visual examination of its outputs. Our work is also an early exploration of how to qualitatively and quantitatively assess methods for interactive GAN exploration.

Our results encourage further exploration in this space. In particular, our work shows the value of tools that support exploration of hard to get search spaces. Future work should therefore explore other tools based on principled, mathematical methods to explore GANs using the working image gallery paradigm. Also, in our work, we have focused on individual categories of images in the BigGAN model as a preliminary investigation. However, models such as BigGAN enable creative generation of images from multiple categories using the principle of Visual Indeterminacy [15]. Therefore, in the future work, we plan to explore this aspect of the BigGAN model in our interactive tool.

Finally, in this work we have explored only a single measure of image quality: photo-realism. However, other GAN output quality measures may be required for more advanced use case scenarios (e.g., for model designers as an interactive diagnostics tool). Also, as GANs become more common design and art support tools, future work will have to explore other measures of quality to inform design of future creativity support tools.

ACKNOWLEDGMENTS

We thank members of the CompHCI Lab at the University of Michigan for their feedback on this work, and in particular Divya Ramesh who provided us with valuable information on design and execution of our MTurk user studies.

REFERENCES

[1] Amazon. 2020. *Amazon Mechanical Turk.* https://www.mturk.com/

[2] Christophe Andrieu, Nando De Freitas, Arnaud Doucet, and Michael I Jordan. 2003. An introduction to MCMC for machine learning. *Machine learning* 50, 1-2 (2003), 5–43.

[3] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Bolei Zhou, Joshua B. Tenenbaum, William T. Freeman, and Antonio Torralba. 2019. GAN Dissection: Visualizing and Understanding Generative Adversarial Networks. In *Proceedings of the International Conference on Learning Representations (ICLR).*

[4] Ali Borji. 2019. Pros and cons of GAN evaluation measures. *Computer Vision and Image Understanding* 179 (2019), 41 – 65. https://doi.org/10.1016/j.cviu.2018.10.009

[5] Eric Brochu, Vlad M Cora, and Nando De Freitas. 2010. A tutorial on Bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. *arXiv preprint arXiv:1012.2599* (2010).

[6] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2018. Large Scale GAN Training for High Fidelity Natural Image Synthesis. arXiv:1809.11096 [cs.LG]

[7] Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large Scale GAN Training for High Fidelity Natural Image Synthesis. In *International Conference on Learning Representations.* https://openreview.net/forum?id=B1xsqj09Fm

[8] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. 2016. Neural photo editing with introspective adversarial networks. *arXiv preprint arXiv:1609.07093* (2016).

[9] Steve Brooks, Andrew Gelman, Galin Jones, and Xiao-Li Meng (Eds.). 2011. *Handbook of Markov Chain Monte Carlo.* Chapman and Hall.

[10] Wengling Chen and James Hays. 2018. SketchyGAN: Towards Diverse and Realistic Sketch to Image Synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

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

[12] Brochu Eric, Nando D Freitas, and Abhijeet Ghosh. 2008. Active preference learning with discrete choice data. In *Advances in neural information processing systems.* 409–416.

[13] Edoardo Giacomello, Pier Luca Lanzi, and Daniele Loiacono. 2019. Searching the Latent Space of a Generative Adversarial Network to Generate DOOM Levels. In *2019 IEEE Conference on Games (CoG).* 1–8. https://doi.org/10.1109/CIG.2019.8848011

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.). Curran Associates, Inc., 2672–2680. http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[15] Aaron Hertzmann. 2020. Visual Indeterminacy in GAN Art. *Leonardo* 53, 4 (2020), 424–428. https://doi.org/10.1162/leon_a_01930 arXiv:https://doi.org/10.1162/leon$_{a0}$1930

[16] Matthew D Hoffman and Andrew Gelman. 2014. The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* 15, 1 (2014), 1593–1623.

[17] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. 2018. Multimodal Unsupervised Image-to-image Translation. In *Proceedings of the European Conference on Computer Vision (ECCV).*

[18] Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Advances in Neural Information Processing Systems*, Vol. 33. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2020/file/6fe43269967adbb64ec6149852b5cc3e-Paper.pdf

[19] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-To-Image Translation With Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

[20] Mario Klingemann. 2020. Trapping the Accident. In *PhotographyDigitalPainting: Expanding Medium Interconnectivity in Contemporary Visual Art Practices* (first ed.), Carl Robinson (Ed.). Cambridge Scholars Publishing, 77–98.

[21] Yuki Koyama, Issei Sato, and Masataka Goto. 2020. Sequential Gallery for Interactive Visual Design Optimization. *ACM Trans. Graph.* 39, 4, Article 88 (July 2020), 12 pages. https://doi.org/10.1145/3386569.3392444

[22] Yuki Koyama, Issei Sato, Daisuke Sakamoto, and Takeo Igarashi. 2017. Sequential Line Search for Efficient Visual Design Optimization by Crowds. *ACM Trans. Graph.* 36, 4, Article 48 (July 2017), 11 pages. https://doi.org/10.1145/3072959.3073598

[23] J. Marks, B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, T. Kang, B. Mirtich, H. Pfister, W. Ruml, K. Ryall, J. Seims, and S. Shieber. 1997. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '97).* ACM Press/Addison-Wesley Publishing Co., USA, 389–400. https://doi.org/10.1145/258734.258887

[24] Jakob Nielsen and Rolf Molich. 1990. Heuristic Evaluation of User Interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seattle, Washington, USA) *(CHI '90).* Association for Computing Machinery, New York, NY, USA, 249–256. https://doi.org/10.1145/97243.97281

[25] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. 2019. Semantic Image Synthesis With Spatially-Adaptive Normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR).*

[26] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. arXiv:1511.06434 [cs.LG]

[27] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved Techniques for Training GANs. In *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett (Eds.). Curran Associates, Inc., 2234–2242. http://papers.nips.cc/paper/6125-improved-techniques-for-training-gans.pdf

[28] John Salvatier, Thomas V Wiecki, and Christopher Fonnesbeck. 2016. Probabilistic programming in Python using PyMC3. *PeerJ Computer Science* 2 (2016), e55.

[29] Helena Sarin. 2019. *The Books of GANesis: Divine Comedy in Tangled Representations.* Helena Sarin.

[30] Jacob Schrum, Jake Gutierrez, Vanessa Volz, Jialin Liu, Simon Lucas, and Sebastian Risi. 2020. Interactive Evolution and Exploration within Latent Level-Design Space of Generative Adversarial Networks. In *Proceedings of the 2020 Genetic and Evolutionary Computation Conference* (Cancún, Mexico) *(GECCO '20).* Association for Computing Machinery, New York, NY, USA, 148–156. https://doi.org/10.1145/3377930.3389821

[31] Ben Shneiderman. 2007. Creativity Support Tools: Accelerating Discovery and Innovation. *Commun. ACM* 50, 12 (Dec. 2007), 20–32. https://doi.org/10.1145/1323688.1323689

[32] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Vancouver, BC, Canada) *(CHI '11).* Association for Computing Machinery, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[33] Huikai Wu, Shuai Zheng, Junge Zhang, and Kaiqi Huang. 2019. GP-GAN: Towards Realistic High-Resolution Image Blending. In *Proceedings of the 27th ACM International Conference on Multimedia* (Nice, France) *(MM '19).* Association for Computing Machinery, New York, NY, USA, 2487–2495. https://doi.org/10.1145/3343031.3350944

[34] Sharon Zhou, Mitchell Gordon, Ranjay Krishna, Austin Narcomey, Li F Fei-Fei, and Michael Bernstein. 2019. HYPE: A Benchmark for Human eYe Perceptual Evaluation of Generative Models. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.). Curran Associates, Inc., 3449–3461. http://papers.nips.cc/paper/8605-hype-a-benchmark-for-human-eye-perceptual-evaluation-of-generative-models.pdf

[35] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. 2016. Generative Visual Manipulation on the Natural Image Manifold. In *Computer Vision – ECCV 2016*, Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling (Eds.). Springer International Publishing, Cham, 597–613.