Figure 1: We develop a search algorithm for customized and pre-trained deep generative models. We collect a diverse set of models, such as animals, landscapes, human faces, and art pieces. From left to right, our search algorithm enables queries in four modalities: text, images, sketches, and existing models. The 1st row consists of the queries, and the 2nd and 3rd rows show the first- and second-ranked models by our method, respectively. Our method succeeds in finding relevant models with similar semantic concepts across all four modalities. Different colors denote different model types. Photograph by @12019 on Pixabay.
ABSTRACT
The growing proliferation of customized and pretrained generative models has made it infeasible for a user to be fully cognizant of every model in existence. To address this need, we introduce the task of content-based model search: given a query and a large set of generative models, finding the models that best match the query. As each generative model produces a distribution of images, we formulate the search task as an optimization problem to select the model with the highest probability of generating similar content as the query. We introduce a formulation to approximate this probability given the query from different modalities, e.g., image, sketch, and text. Furthermore, we propose a contrastive learning framework for model retrieval, which learns to adapt features for various query modalities. We demonstrate that our method outperforms several baselines on Generative Model Zoo, a new benchmark we create for the model retrieval task.

CCS CONCEPTS
• Computing methodologies → Artificial intelligence; Machine learning.

KEYWORDS
Imaging & Video, Machine Learning, Artificial Intelligence

ACM Reference Format:
Daohan Lu, Sheng-Yu Wang, Nupur Kumari, Rohan Agarwal, Mia Tang, David Bau, and Jun-Yan Zhu. 2023. Content-Based Search for Deep Generative Models. In SIGGRAPH Asia 2023 Conference Papers (SA Conference Papers ’23), December 12–15, 2023, Sydney, NSW, Australia. ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3610548.3618189

1 INTRODUCTION
We introduce the task of content-based model search, which aims to find the most relevant deep image generative models that satisfy a user’s input query. For example, as shown in Figure 1, we enable a user to retrieve a model capable of synthesizing images that match a text query (e.g., Miniatures), an image query (e.g., a landscape photo), sketch query (e.g., Mario sketch), or models similar to a given model. Users can leverage retrieved models in various ways: they might further fine-tune it with a unique concept, use the model to generate an image with a more concrete prompt, or combine multiple retrieved models into one.

Content-based model search is becoming a vital task as the number and diversity of intriguing generative models continue to explode. These models have formed a new form of media content that requires efficient indexing and searching. While one might have expected large-scale text-to-image models [Ramesh et al. 2022; Rombach et al. 2022; Saharia et al. 2022] to have led to a shrinking universe of useful models, instead, this powerful class of content generators has touched off an accelerating proliferation of new and customized models. That is because powerful generative models are capable of being adapted to capture a wide range of personalized subjects [Gal et al. 2023; Kawar et al. 2023; Kumari et al. 2023; Ruiz et al. 2022]. Everyday users are routinely creating new generative models, and the community has collectively shared tens of thousands of custom models on community-driven platforms, such as Civitai [civ 2022] and HuggingFace [hug 2022] in the past year alone. Many popular models arise from the model creators’ unique creative processes, involving careful selections of subjects, algorithms, hyperparameters, and often proprietary artistic training data. Furthermore, some model creators receive compensation when other users download or use their models.

Aside from personalized models, a variety of deep generative models are being created as backbones for computer graphics applications [Bermano et al. 2022; Tewari et al. 2020], and as works of art that explore a wide range of themes [Elgammal 2019; Hertzmann 2020]. Each model captures a small universe of curated subjects, which can range from the realistic rendering of faces and landscapes [Karras et al. 2021] to photos of historic pottery [Au 2019] to cartoon caricatures [Jang et al. 2021] to single-artist stylistic elements [Schultz 2020]. Various methods also enable creative modifications of existing models via human-in-the-loop interfaces [Bau et al. 2020; Gal et al. 2022b; Wang et al. 2022].

Given the accelerating pace at which generative models are being created and uploaded to the internet, the ability to conduct content-based model search will be instrumental in supporting the effective use of large model collections. Existing model-sharing platforms [civ 2022; hug 2022] primarily rely on matching human-created model names for model search. However, it is difficult to choose sufficiently detailed names to describe the unique, complex, and highly specific images produced by a generative model. Our work aims to define a new approach, searching for models directly based on their content rather than based on manual definitions alone.

Content-based model search is a challenging task: even the simplified question of whether a specific image can be produced by a single model can be computationally difficult. Unfortunately, many generative models do not offer an efficient or exact way to estimate density, nor do they natively support assessing cross-modal similarity (e.g., text and image).

To address the above challenges, we first curate a benchmark retrieval dataset, the Generative Model Zoo, consisting of (1) 259 publicly available generative models that vary in content as well as model architectures, including GANs (e.g., StyleGAN-family models [Karras et al. 2020b]), diffusion models (e.g., DreamBooth [Ruiz et al. 2022]), and auto-regressive models (e.g., VQGAN [Esser et al. 2021]) and (2) 1000 customized text-to-image diffusion models, each one fine-tuned on a single instantiation of an object class or a single artistic image. As part of our benchmark, we define ground truth image, text, and sketch queries for evaluating model retrieval.

We present a general probabilistic formulation of the model search problem and propose a learning-based method given this formulation. We summarize our contribution as follows:

• We introduce a new task of content-based search over deep generative models. Given a piece of text, an image, a sketch,
a generative model, or a combination of them as a query, we aim to return the most relevant generative models that can synthesize similar content.

- We formulate the task of content-based model retrieval as estimating the probability of generating an image that matches the query content. We propose a new contrastive learning method to estimate the same given the model’s image distribution statistics and query. Our learning-based method outperforms several baseline algorithms.
- We curate a benchmark dataset, the Generative Model Zoo, which includes a diverse compilation of more than 250 community-created generative models, 1000 single-image fine-tuned models, and a set of ground truth (query, model) pairs for evaluating the retrieval algorithm on different query modalities.

2 RELATED WORKS

Deep generative models. Generative models are open-sourced at a rate of thousands per month. They use different learning objectives [Goodfellow et al. 2014; Ho et al. 2020; Kingma and Welling 2014; Oord et al. 2016; Song et al. 2021b], training techniques [Karras et al. 2020a; Mokady et al. 2022; Rombach et al. 2022; Sauer et al. 2021], and network architectures [Brock et al. 2019; Esser et al. 2021; Karras et al. 2019; Razavi et al. 2019]. They are also trained on different datasets [Choi et al. 2020; Mokady et al. 2022; Schultz 2020; Yu et al. 2015] for different applications [Albahar et al. 2021; Chen et al. 2020; Ha and Schmidhuber 2018; Lewis et al. 2021; Patashnik et al. 2021; Razavi et al. 2021; Zhang et al. 2021; Zhu et al. 2021]. This diversity leads to the question, among all the models, which one shall we use? Our goal is not to introduce a new model. Instead, we want to help researchers, students, and artists find existing models more easily.

Model editing and fine-tuning. As methods for editing and fine-tuning generative models become more accessible and efficient, these have also contributed to a proliferation of models. Several works edit a pre-trained generative model with simple user interfaces like sketching [Wang et al. 2021], warping [Wang et al. 2022], blending [Bau et al. 2020], and text prompts [Gal et al. 2022b]. Besides editing a model, a line of works propose methods to fine-tune models to match a small collection of images [Karras et al. 2020a; Li et al. 2020; Mo et al. 2020; Nitzan et al. 2022; Noguchi and Harada 2019; Ojha et al. 2021; Wang et al. 2020, 2018; Zhao et al. 2020a,b]. Various works further improve the fine-tuning speed [Grigorov et al. 2022; Kumari et al. 2022; Liu et al. 2021; Sauer et al. 2021].

Customization of large-scale diffusion models. Recently, large-scale text-to-image models [Ramesh et al. 2022; Rombach et al. 2022; Saharia et al. 2022] have shown exemplary performance in generating diverse styles and compositions given the text prompt, with downstream versatile image editing capacity [Avrahami et al. 2022; Brooks et al. 2023; Hertz et al. 2022; Tumanyan et al. 2023; Zhang et al. 2023]. But these models cannot still synthesize personalized and artistic concepts that are hard to describe via text. To enable that, various works have proposed model customization techniques [Gal et al. 2022a; Han et al. 2023; Kawar et al. 2023; Kumari et al. 2023; Ruiz et al. 2022]. This has given rise to tens of thousands of fine-tuned models for different styles and concepts [civ 2022; hug 2022], thus making model search increasingly relevant. Recently, several works use retrieval-augmented generative models to improve the fidelity of rare entities [Blattmann et al. 2022; Casanova et al. 2021; Chen et al. 2023; Ma et al. 2023]. Different from these, our goal is to simply find a model relevant to user queries, rather than generating an exact image given the query.

Content-based retrieval. Building upon classical information retrieval [Baeza-Yates et al. 1999; Manning et al. 2010], content-based retrieval deals with queries over image, video, or other media [Datta et al. 2008; Gudivada and Raghavan 1995; Hu et al. 2011]. Content-based image retrieval methods use robust visual descriptors [Dalal and Triggs 2005; Lowe 2004; Oliva and Torralba 2001] to match objects within video or images [Arandjelović and Zisserman 2012; Sivic and Zisserman 2003]. Methods have been developed to compress visual features to scale retrieval [Gong et al. 2012; Jégou et al. 2010; Torralba et al. 2008; Weiss et al. 2008], and deep learning has enabled compact vector representations for retrieval [Babenko et al. 2014; Krizhevsky and Hinton 2011; Torralba et al. 2008; Zheng et al. 2017]. In addition to image queries, various works have proposed methods for sketch-based retrieval [Eitz et al. 2010; Lin et al. 2013; Liu et al. 2017; Radenovic et al. 2018; Ribeiro et al. 2020; Sangkloy et al. 2016; Yu et al. 2016]. There is a growing interest in joint visual-language embeddings [Faghri et al. 2017; Frome et al. 2013; Jia et al. 2021; Karpathy et al. 2014; Radford et al. 2021; Socher et al. 2014] that enable text queries for image content. We also adopt deep image representations for our setting, but unlike single-image retrieval, we index distributions of images that cannot be fully materialized. Concurrent to our work, HuggingGPT [Shen et al. 2023] also explores model retrieval but for user-defined tasks such as object detection.

3 METHODS

In this section, we develop a retrieval framework for deep generative models. When a user specifies an image, sketch, or text query, we would like to retrieve a model that best matches the query. We denote a collection of $N$ models by $\{\theta_1, \theta_2, \ldots, \theta_N\}$ and the user query by $q$, and we assume a uniform prior over models (i.e., $\theta \sim \text{unif}(\theta_1, \theta_2, \ldots, \theta_N)$). Every model $\theta$ captures a distribution $p(x|\theta)$ over images $x$. While prior retrieval methods [Manning et al. 2010; Smeulders et al. 2000] search for single instances, we aim to construct a method for retrieving distributions.

To achieve this, we introduce a probabilistic formulation for generative model retrieval. Figure 2 shows an overview of our approach. Our formulation is general to different query modalities and various types of generative models, and can be extended to different algorithms. In Section 3.1, we derive our model retrieval formulation based on a Maximum Likelihood Estimation (MLE) objective based on pre-trained deep learning features. In Section 3.2, we further propose a contrastive learning method to adapt features to different query modalities and model search task. We present our model retrieval algorithms for an image, a text, and a sketch query, respectively. In Sections 4 and 5, we discuss our new benchmark and user interface for model search.
3.1 Probabilistic Retrieval for Generative Models

Our goal is to quantify the posterior probability of each model $\theta$ given the user query $q$, and retrieve the model with the maximum $p(\theta|q)$. We define the probabilistic model retrieval objective as:

$$\max_{\theta \in \{\theta_1, \ldots, \theta_N\}} p(\theta|q) = \frac{p(q|\theta)p(\theta)}{p(q)} \propto \max_{\theta \in \{\theta_1, \ldots, \theta_N\}} p(q|\theta). \quad (1)$$

As we assume models are uniformly distributed, it is equivalent to finding the likelihood of the query under each model, $p(q|\theta)$.

There are two scenarios for inferring the query likelihood $p(q|\theta)$:
1. When the query $q$ shares the same modality with the model $\theta$ (e.g., searching models with image queries), we directly reduce the problem to estimating the model’s density.
2. When the query $q$ has a different modality from the model $\theta$ (e.g., searching models with text queries), we use cross-modal similarity to estimate $p(q|\theta)$. We discuss both cases in the following.

### Image-based model retrieval.
Given an image query $q$, we directly estimate the likelihood of the query $p(q|\theta)$ from each model. In other words, we seek to find the model that is most likely to generate the query image. Since density estimation is intractable for many generative models (e.g., GANs [Goodfellow et al. 2014], VAEs [Kingma and Welling 2014]), we approximate each model as a Gaussian distribution over image features [Heusel et al. 2017]. After we sample an image from each model, denoted by $x$, we obtain its image feature $z := \phi_{\text{im}}(x)$, where $\phi_{\text{im}}$ is a frozen feature extractor. We now express $p(q|\theta)$ in terms of image features $z$, so Equation 1 becomes:

$$\max_{n \in \{1, \ldots, N\}} \frac{1}{(2\pi|\Sigma_n|)^{1/2}} \exp\left(-\frac{1}{2}(z_q - \mu_n)^T \Sigma_n^{-1}(z_q - \mu_n)\right). \quad (2)$$

where the query image feature is denoted by $z_q := \phi_{\text{im}}(q)$, and each model $\theta_n$ is approximated by $p(z|\theta_n) \sim \mathcal{N}(\mu_n, \Sigma_n)$. We refer to this method as Gaussian Density.

### Text-based model retrieval.
Given a text query $q$ and a generative model $p(x|\theta)$ capturing a distribution of images $x$, we want to estimate the conditional probability $p(q|\theta)$.

$$p(q|\theta) = \int p(q|x)p(x|\theta) dx \propto \int \frac{p(x|q)}{p(x)} p(x|\theta) dx. \quad (3)$$

Here we assume conditional independence between query $q$ and model $\theta$ given image $x$, so $p(q|x, \theta) = p(q|x)$, and we apply Bayes’ rule to get the final expression. In Equation 3, a text query $q$ may correspond to multiple possible image matches $p(x|q)$, so the expression cannot be simplified the same way as image-based model retrieval. Instead, we can view Equation 3 as an integral of the mutual information between $q$ and $x$, which can be estimated with a contrastive representation [Oord et al. 2018]. Thus, we use CLIP similarity [Radford et al. 2021] to approximate $\frac{p(x|q)}{p(x)} \propto \exp(\frac{1}{\tau}h_q^T h_x)$, where $h_x := \phi_{\text{im}}(x)$ and $h_q := \phi_{\text{txt}}(q)$ are the normalized CLIP image and text features, $\tau$ is a temperature term.

To approximate the integration, one can sample images $x$ from $p(x|\theta)$ and then take the average of the CLIP similarities between images and query. We refer to this method as Monte-Carlo.

To further speed up computation, we can pre-compute the mean of CLIP image embeddings for each model, and at inference time, we directly evaluate similarities between the query embedding and the mean embedding, as follows:

$$\int \frac{p(x|q)}{p(x)} p(x|\theta) dx \approx \exp\left(\frac{\mu_n^T h_q}{\tau}\right), \quad (4)$$

where $\mu_n = E_{h_x \sim p(h_x|\theta_n)} h_x$; $\hat{\mu}_n = \frac{\mu_n}{||\mu_n||}$.
We find that this approximation works well in practice, and we refer to this method as 1st Moment. Derivation details for Monte-Carlo and 1st Moment can be found in the supplement.

1st Moment method for image queries. Likewise, we find that applying 1st Moment for image-based model retrieval yields a performance close to Gaussian Density. Interestingly, for sketch queries, 1st Moment method outperforms Gaussian Density. While CLIP shows strong cross-modal matching performance between sketch queries and image models, the domain gap makes density estimation a less favorable option for sketch queries compared to the 1st Moment method. We provide more discussions in Section 6.

3.2 Learning to Retrieve Models

We have outlined several ways to approximate \( p(\theta | q) \) in Section 3.1 to perform content-based model retrieval. However, these approximations involve only pre-trained feature extractors and may not be optimal for model retrieval with certain query modalities. For example, the frozen features that work well for images may struggle with sketches. To address this issue, we introduce a set of learnable parameters to fine-tune the approximations of \( p(\theta | q) \) on our Generative Model Zoo dataset (Section 4).

Under the contrastive learning framework, we maximize

\[
\mathbb{E} \left[ \log \frac{p(\theta | q)}{p(q)} \right],
\]

the mutual information between the query \( q \) and the model \( \theta \). Since we assume a uniform prior over the models, the mutual information is effectively our retrieval objective \( \mathbb{E} \left[ \log p(\theta | q) \right] \). The training dataset \( D \) consists of sample statistics for \( M \) models and a number of ground-truth query-model pairs. We optimize a matching function \( s_\psi \) that yields a query-to-model similarity score, by minimizing the InfoNCE loss [Oord et al. 2018]:

\[
\min_\psi \mathbb{E} \left[ \log \frac{p(\theta | q)}{p(q)} \right] - \log \frac{\prod_{i=1}^{M} \exp(s_\psi(q, \theta_i)/\tau)}{\sum_{i=1}^{M} \exp(s_\psi(q, \theta_i)/\tau)}.
\]

where \( A_\psi \) is a matrix parameterized by \( \psi \), \( (\mu_\theta, \Sigma_\theta) \) are the feature-space sample mean and covariance of model \( \theta \), and \( \tilde{h}_\theta \) is as calculated in the previous section using a pre-trained feature extractor. The contrastive learning objective maximizes the similarity score between query \( q \) and the ground truth model and minimizes the similarity between the query and other models.

We define the matching function based on two formulations described in Section 3.1. 1st Moment and Gaussian Density. For 1st Moment, we project the original feature using the matrix \( A_\psi \). For Gaussian Density, \( \mathcal{N}(\tilde{h}_\theta | \mu_\theta, A_\psi \Sigma_\theta A_\psi^T) \) denotes the gaussian PDF with parameters \( (\mu_\theta, A_\psi \Sigma_\theta A_\psi^T) \), evaluated at the query feature \( \tilde{h}_\theta \). We experiment with different \( A_\psi \) parameterizations, including full, triangular, and diagonal matrices. For the diagonal case, we limit the range of \( A_\psi \) such that \( A_\psi = \text{diag}(\text{sigmoid}(\psi)) \).

In Section 6, we show that our method can retrieve models that share similar visual concepts with the query.

4 THE GENERATIVE MODEL ZOO

We introduce a new benchmark, the Generative Model Zoo, consisting of a set of generative models capturing a variety of architectures and subject areas for evaluating retrieval performance.

Internet model zoo. It consists of a total of 259 publicly available generative models trained using different techniques, including GANs [Gal et al. 2022b; Karras et al. 2018, 2021, 2020b; Kumari et al. 2022; lucid layers 2022; Mokady et al. 2022; Pinkney 2020; Sauer et al. 2022; Wang et al. 2021, 2022], diffusion models [Dhariwal and Nichol 2021; Ho et al. 2020; Ruiz et al. 2022; Song et al. 2021a], MLP-based generative model CIPS [Anokhin et al. 2021], and the autoregressive model VQGAN [Esser et al. 2021]. Out of the 259 models, 23 were created via fine-tuning by individual users, 133 were obtained from the GitHub repositories of academic papers, and the remaining models were collected from public model sharing website [hug 2022]. This includes models like DreamBooth [Ruiz et al. 2022] that are finetuned to generate a specific concept (e.g., a unique toy) with a fixed text prompt as input. Hence, to index these models, we generate samples conditioned on the unique text prompt used to fine-tune the model. A comprehensive list of the models and their respective sources is included in the supplement.

We create a set of benchmark queries for each model in multiple modalities, i.e., text, image, and sketch. The text queries include human-written textual queries that describe the images in a model. Image queries are created by sampling images from models, and sketch queries are created by using the method of Chan et al. [2022].
Synthetic model zoo. To further test our method at scale, we create a synthetic model zoo of 1,000 models. Each model is finetuned from the pretrained Stable Diffusion v1.4 checkpoint [Rombach et al. 2022] on a single image. The models are fine-tuned on instances from animal classes in ImageNet [Deng et al. 2009] or artistic images downloaded from Unsplash [Unsplash 2022]. We manually pick 50 ImageNet classes and 10 instance images from each class, thus training 500 models with instance images from the ImageNet Dataset. For the artistic models, we select 500 images from Unsplash that match the keyword “art”. To fine-tune the model, we randomly select between Dreambooth [Ruiz et al. 2022] with LoRA [Hu et al. 2021] and Custom Diffusion [Kumari et al. 2023]. In this case, we only create image and sketch queries, as describing specific instantiation of a general category from ImageNet classes is difficult via only text. Similarly, describing an art work using text is also non-trivial.

5 USER INTERFACE

Our baseline method is fast enough for interactive use, and we create a web-based UI for the search algorithm. The user can search for models by entering text or uploading an image/sketch. The interface displays models that best match the query, where clicking on a model’s thumbnail shows more image samples. The website utilizes a backend GPU server to enable real-time model search on any client device. Figure 3 shows a screenshot of our UI.

6 EXPERIMENTS

For our baseline retrieval methods, we benchmark their performance over text, image, and sketch modalities and discuss several algorithmic design choices.

Implementation details. We test our method with the commonly used pretrained feature extractor – CLIP [Radford et al. 2021]. We use the $l_2$-normalized CLIP features, as discussed in Section 3.1.

We pre-calculate each model’s generated distribution mean and covariance in the CLIP feature space, following the pre-processing steps described in c1eann-f1d [Parmar et al. 2022]. For ImageNet fine-tuned models, we generate class-specific prompts used to sample images using ChatGPT. For artistic fine-tuned models, we use general prompts, e.g., “a painting in the style of <specific art style>”. More implementation details are in the supplement.

To evaluate model search, we use the top-k accuracy metric, i.e., frequency of finding the unique ground truth model within the top-k retrieved models using the query corresponding to that model.

6.1 Model Retrieval

Model retrieval via image and sketch queries. We report top-k accuracy of model retrieval on the Internet model zoo dataset in Table 1. We show the performance of different formulations with CLIP features. We train the transformation matrix $A_\psi$ in the learning to retrieve method using the Synthetic model zoo. The retrieval performance is best with CLIP features combined with our learning method, especially for sketch-based retrieval. The learning method is denoted with the suffix (FT) in Table 1. For comparison, we also include a modified k-NN algorithm in which we find the query’s feature-space k-nearest neighbors among each model’s image samples and sort the models by the mean distance to those k samples. We refer to this baseline as “best-k neighbors” or simply “best-k”.

Table 1: Image- and sketch-based model retrieval. We compare retrieval performance with pre-trained and fine-tuned scoring functions utilizing CLIP network backbone. We observe the best performance with CLIP and fine-tuning. The (FT) suffix denotes the learning-based method (Equation 6) using the optimal parameterization selected via first doing cross-validation on the Synthetic Model Zoo as shown in Table 2.

| Modality | top-k Accuracy | Mean Testing Accuracy |
|----------|---------------|-----------------------|
|          | Top-1         | Top-5                 | Top-10                |
| CLIP+Monte-Carlo (2.4K) | 0.82 | **0.96** | 0.99 |
| CLIP+best-k (2.4K, k=1) | 0.80 | 0.95 | 0.99 |
| CLIP+best-k (2.4K, k=10) | 0.82 | **0.96** | 0.99 |
| CLIP+1st Moment | 0.78 | 0.94 | 0.98 |
| CLIP+1st Moment (FT) | 0.81 | 0.95 | 0.98 |
| CLIP+Gaussian Density | 0.84 | **0.96** | 0.98 |
| CLIP+Gaussian Density (FT) | 0.85 | **0.96** | 0.99 |
| CLIP+Monte-Carlo (2.4K) | 0.32 | 0.59 | 0.72 |
| CLIP+best-k (2.4K, k=1) | 0.28 | 0.55 | 0.68 |
| CLIP+best-k (2.4K, k=10) | 0.32 | 0.59 | 0.72 |
| CLIP+1st Moment | 0.32 | 0.61 | 0.73 |
| CLIP+1st Moment (FT) | 0.49 | **0.75** | **0.84** |
| CLIP+Gaussian Density | 0.18 | 0.39 | 0.53 |
| CLIP+Gaussian Density (FT) | 0.30 | 0.56 | 0.69 |

Table 2: Selecting best parameterization for $A_\psi$ with cross-validation. We fine-tune different versions of similarity function and transformation matrix parameterization with 5-fold cross-validation on the Synthetic Model Zoo. We use it to determine the optimal parameterization of $A_\psi$ (marked with a “★”) for each method and modality.

| Modality | Mean Testing Accuracy |
|----------|-----------------------|
|          | Top-1 | Top-5 |
| CLIP+1st Moment | Image | Diagonal | 0.84 | 0.81 |
| CLIP+1st Moment | Image | Triangular | 0.90 | 0.81 |
| CLIP+1st Moment | Image | Full | 0.89 | 0.81 |
| CLIP+Gaussian Density | Image | Diagonal | 0.91 | 0.88 |
| CLIP+Gaussian Density | Image | Triangular | 0.91 | 0.88 |
| CLIP+Gaussian Density | Image | Full | 0.91 | 0.88 |
| CLIP+1st Moment | Sketch | Diagonal | 0.30 | 0.15 |
| CLIP+1st Moment | Sketch | Triangular | 0.61 | 0.15 |
| CLIP+1st Moment | Sketch | Full | 0.61 | 0.15 |
| CLIP+Gaussian Density | Sketch | Diagonal | 0.29 | 0.09 |
| CLIP+Gaussian Density | Sketch | Triangular | 0.51 | 0.09 |
| CLIP+Gaussian Density | Sketch | Full | 0.38 | 0.09 |
Table 3: Learning to retrieve with text queries. We show the result of the learning-based method with text queries using a 5-fold cross-validation on the Internet Model Zoo itself. This is because we have ground truth text queries only for that, as discussed in Section 4. We optimize different parameterizations of the transformation matrix on the training split and show retrieval performance on the testing split of 51 models.

| Method             | Modality | Top-1 (FT) | Top-1 (Pre-trained) |
|--------------------|----------|------------|---------------------|
| CLIP+Monte-Carlo (2.4K) | Text     | -          | 0.75                |
| CLIP+best-k (2.4K, k=1) | Text     | -          | 0.67                |
| CLIP+best-k (2.4K, k=10) | Text     | -          | 0.74                |
| CLIP+1st Moment | Text      | Diagonal   | 0.78                |
| CLIP+1st Moment | Text      | Triangular | 0.80                |
| CLIP+1st Moment | Text      | Full       | 0.81                |

As shown in Table 4, the content-based search methods outperform the baselines while they also obviate the need for model descriptions. Metadata-based search typically fails when captions are incorrect or when the object-centric tags cannot describe the model fully. Creating comprehensive descriptions might reduce the gap, but it is time-consuming to describe every visual aspect and anticipate users’ queries in advance. Our evaluation does not include text-based queries, as we use the model descriptions as the ground truth. However, we expect the baselines to have the same limitation for text queries in the case where a text query requests a visual concept that the tags fail to enumerate.

Running time and memory. Time and memory efficiency are crucial for supporting many concurrent user searches over a large-scale model collection. While 1st Moment obtains competitive top-k accuracies, it runs 3.2-7.5 times faster than Monte-Carlo or Gaussian Density. The fine-tuned 1st Moment method further improves accuracy, but runs at speeds more on par with Monte-Carlo or Gaussian Density. 1st Moment is extremely memory-efficient. Further analysis is in the supplement.

7 EXTENSIONS

We describe two extensions: searching with multimodal queries and searching with a given model.

Multimodal user query. We show qualitatively that our search method can be extended to multimodal queries based on the Product-of-Experts formulation [Hinton 2002; Huang et al. 2022]. Given a multimodal query (e.g., text-image pair), the retrieval score is the product of individual query likelihoods. We demonstrate how one can leverage multiple input modalities to perform nuanced searches in Figure 4 using the 1st Moment method with pre-trained features. We show additional multimodal retrieval results in the supplement.

Generative model query. Given the generative model collection, we can also retrieve similar models based on the cosine similarity between the feature-space means of query models and other generative models. Figure 5 shows qualitative examples of similar-model retrieval using CLIP feature space.

8 DISCUSSION AND LIMITATIONS

We have introduced the new task of content-based generative model search. We have developed a data set for benchmarking model retrieval algorithms, and we have described several baselines and a contrastive learning method for further improving features.
Table 4: Comparison with metadata-based search. We compare our Gaussian Density content-based search method with two metadata-based search methods: (1) Description (text): we caption image/sketch queries using BLIP [Li et al. 2022] and select models whose descriptions contain any non-filler words from the caption, (2) Description (CLIP), we embed model descriptions into the CLIP feature space and compare the similarity between the CLIP query features and the CLIP description features.

|       | Top-1 | Top-5 | Top-10 |
|-------|-------|-------|--------|
| Image (Gen.) |       |       |        |
| Description (text) | 0.07  | 0.20  | 0.29   |
| Description (CLIP)  | 0.45  | 0.72  | 0.85   |
| CLIP+Gaussian Density (ours) | **0.89** | **0.98** | **0.99** |
| Sketch |       |       |        |
| Description (text) | 0.07  | 0.20  | 0.28   |
| Description (CLIP)  | 0.22  | 0.46  | 0.60   |
| CLIP+1st Moment (ours) | **0.49** | **0.75** | **0.84** |

Figure 4: Multi-modal user query. Image and text queries combined can define nuanced concepts. When querying with the phrase “pointy eyes” plus a cat photo, we can retrieve “Triangle Eyes Cat” and “Alien Eyes Cat”. Photo by cocoparisienne on Pixabay.

Figure 5: Finding similar models. Our method can find models that share similar characteristics with the query model.

Limitations. While we can retrieve models in real-time based on text, image, or sketch queries, our method has several limitations.

First, while our method is able to successfully retrieve models that capture a single concept (e.g., StyleGAN3 [Karras et al. 2021] and DreamBooth [Ruiz et al. 2022]), there remain many future directions, including searching for personalized models that contain multiple subjects [Kumari et al. 2023], 3D neural objects [Mildenhall et al. 2021; Poole et al. 2022], and other media such as text, audio, and videos.

Second, our method can sometimes fail to match the user intent. A typical failure case is illustrated in Figure 6, where a query for a sketch of left facing horse retrieves general horse models instead of the most similar GANSketch “Left-facing Horse” model. It fails to respect the spatial orientation of the object in the query image. We show further analysis on this in the supplement.

Nevertheless, our experiments have shown that it is realistic, possible, and useful to search an indexed collection of models by matching them against the output behavior of the models. As the number of customized and pre-trained models continues to balloon, we anticipate that search algorithms that can effectively locate relevant customized models will be an increasingly valuable resource for researchers and content creators.

ACKNOWLEDGMENTS

We thank George Cazenavette, Gaurav Parmar, Chonghyuk (Andrew) Song, and Or Patashnik for proofreading the draft. We are also grateful to Aaron Hertzmann, Maneesh Agrawala, Stefano Ermon, Chenlin Meng, William Peebles, Richard Zhang, Phillip Isola, Taesung Park, Muyang Li, Kangjie Deng, and Ji Lin for helpful comments and discussion. The work is partly supported by NSF IIS-2239076, Adobe Inc., Sony Corporation, and Naver Corporation.

REFERENCES

2022. Civit AI. https://civitai.com.
2022. Stable Diffusion Dreambooth Concepts Library. https://huggingface.co/sd-dreambooth-library.
Badour Albahar, Jingwan Lu, Jimei Yang, Zhixin Shu, Eli Shechtman, and Jia-Bin Huang. 2021. Pose with Style: Detail-Preserving Pose-Guided Image Synthesis with Conditional StyleGAN. ACM TOG (2021).
Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempiński, and Denis Korzhenkov. 2021. Image Generators with Conditionally-Independent Pixel Synthesis. In CVPR.
Relja Arandjelović and Andrew Zisserman. 2012. Three things everyone should know to improve object retrieval. In CVPR.
Derek Philip Au. 2019. This vessel does not exist. https://thisvesseldoesnotexist.com/.
Omri Avrahami, Dani Lischinski, and Ohad Fried. 2022. Blended diffusion for text-driven editing of natural images. In CVPR.
Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. 2023. Plug-and-play diffusion features for text-driven image-to-image translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 1921–1930. Unsplash. 2022. Unsplash. https://unsplash.com.

Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. 2021. Sketch Your Own GAN. In ICCV. Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. 2022. Rewriting Geographic Rules of a GAN. ACM TOG (2022).

Yaxing Wang, Abel Gonzalez-Garcia, David Berga, Luis Herranz, Fahad Shahbaz Khan, and Joost van de Weijer. 2020. Minegan: effective knowledge transfer from gans to target domains with few images. In CVPR.

Yaxing Wang, Chenchen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. 2018. Transferring gans: generating images from limited data. In ECCV.

Yair Weiss, Antonio Torralba, and Rob Fergus. 2008. Spectral hashing. In NIPS.

Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015).

Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. 2016. Sketch me that shoe. In CVPR.

Lvmin Zhang, Anyi Rao, and Manesh Agrawala. 2023. Adding conditional control to text-to-image diffusion models. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 3836–3847.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-François Lalonde, Adela Barriuso, Antonio Torralba, and Sanja Fidler. 2021. Datasetgan: Efficient labeled data factory with minimal human effort. In CVPR.

Miaoyan Zhao, Yulai Cong, and Lawrence Carin. 2020a. On leveraging pretrained GANs for generation with limited data. In ICLR.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. 2020b. Differentiable Augmentation for Data-Efficient GAN Training. In NeurIPS.

Liang Zheng, Yi Yang, and Qi Tian. 2017. SIFT meets CNN: A decade survey of instance retrieval. IEEE TPAMI (2017).

Peihao Zhu, Rameen Abdal, John Fenneri, and Peter Wonka. 2021. Barbershop: Gan-based image compositing using segmentation masks. ACM TOG (2021).
Figure 7: Qualitative results of model retrieval. Top row (image query): Our algorithm retrieves matching models of indoor scenes and landscapes. Middle row (sketch query): Animal sketches retrieve other animals, and more simplistic sketches (e.g. the cat sketch) tend to retrieve cartoonish models. Bottom row (text query): We retrieve relevant models matching the textual description. Photos by Stuart Bailey and @timbri97 on Pixabay. Sketches by Damaris Dharmah and @OpenClipart-Vectors on Pixabay.
Figure 8: Qualitative results of model retrieval on Synthetic Model Zoo. Top row (image query): Our algorithm retrieves models of the dog breed Great Pyrenees and artistic flowers, given matching image queries. Middle row (sketch query): Given sketches of a fruit basket and a bird, our algorithm retrieves models that generate a collection of fruits, and other birds, respectively. Bottom row (text query): We can also retrieve relevant models with a textual description. Images by @drcaringola and @Owantana on Pixabay. Sketches by @N-region and @Tilixia on Pixabay.