Content-Based Search for Deep Generative Models

ANONYMOUS AUTHOR(S)

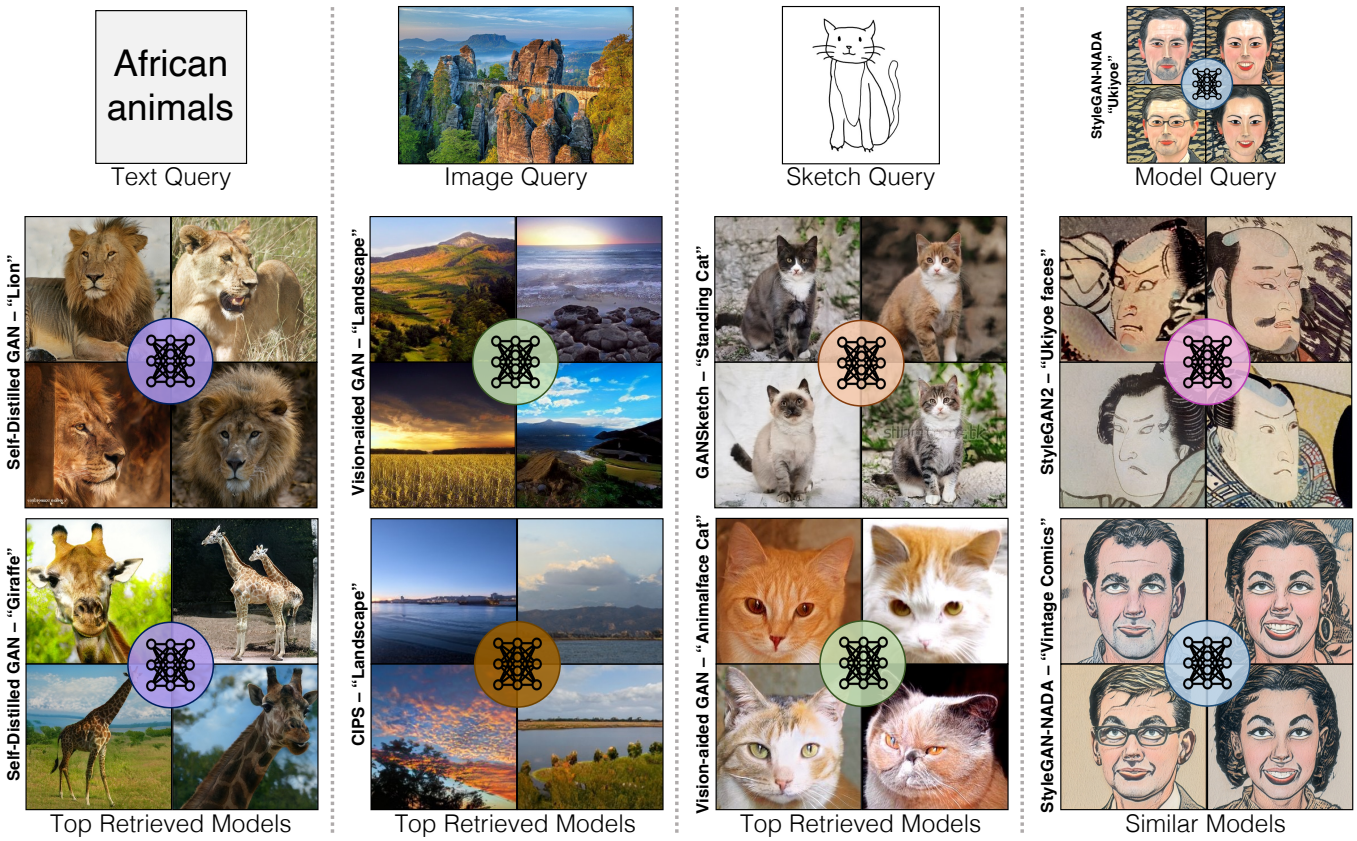

Fig. 1. We develop a search system for deep generative models. We collect a diverse set of models, such as animals, landscapes, human faces, and art pieces. From left to right, our search algorithm enables queries with four different modalities – text, images, sketches, and existing models. The 1st row consists of the queries, and the 2nd and 3rd rows show the first- and second-ranked model by our method, respectively. The color of each model icon implies the generative model type. Our method succeeds in finding relevant models with similar semantic concepts in all four modalities.

The growing proliferation of pretrained generative models has made it infeasible for a user to be fully cognizant of every model that exists. To address this problem, we introduce the problem of *content-based model retrieval*: given a query and a large set of generative models, finding the models that best match the query. Because each generative model produces a distribution of images, we formulate the search problem as an optimization to maximize the probability of generating a query match given a model. We develop approximations to make this problem tractable when the query is an image, a sketch, a text description, another generative model, or a combination of the above. We benchmark our methods in both accuracy and speed over a set of generative models. We further demonstrate that our model search can retrieve good models for image editing and reconstruction, few-shot transfer learning, and latent space interpolation.

0730-0301/2022/9-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

ACM Reference Format:
Anonymous Author(s). 2022. Content-Based Search for Deep Generative Models. *ACM Trans. Graph.* 1, 1 (September 2022), 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

We present the task of content-based model retrieval, which aims to find the most relevant deep image generative models that satisfy a user's input query. For example, as seen in Figure 1, we enable a user to retrieve a generative model either based on its ability to synthesize images that match an image query (e.g., a landscape photo), text query (e.g., `African animals`), or sketch query (e.g., sketch of a standing cat) or its similarity to a given query model.

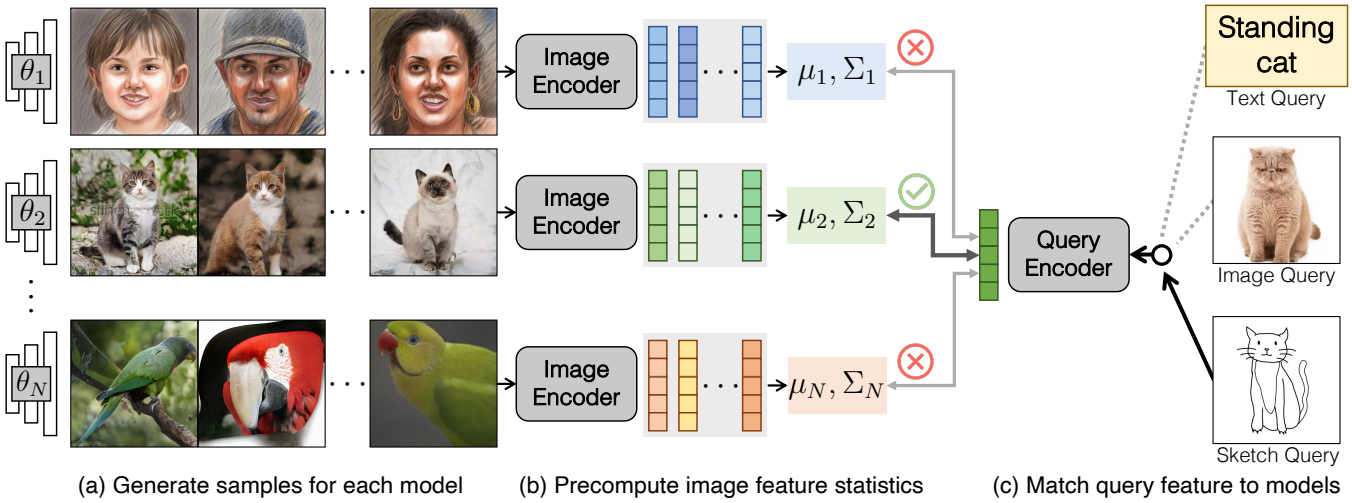

Fig. 2. **Method Overview.** Our search system consists of a pre-caching stage (a, b) and an inference stage (c). Given a collection of models $\theta \sim \text{unif}\{\theta_1, \theta_2, \ldots, \theta_N\}$, (a) we first generate 50K samples for each model θ_n. (b) We then encode the images into image features and compute the 1st and 2nd order feature statistics for each model. The statistics are cached in our system for efficiency. (c) At inference time, we support queries of different modalities (text, image, or sketch). We encode the query into a feature vector, and assess the similarity between the query feature and each model's statistics. The models with the best similarity measures are retrieved. More details of our algorithm are in Section 3.

But why is content-based model search useful? We believe that model search is desperately needed to handle a burgeoning proliferation of generative models: no longer being merely outputs of scientific study, deep generative models are being created as backbones for content creation applications and software [Bermano et al. 2022; Liu et al. 2020; Tewari et al. 2020], as pre-trained models for computer vision and robotics research [Chen et al. 2020; Ha and Schmidhuber 2018], 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 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]. More recently, various methods enable creative modifications of existing generative models, via object compositing [Bau et al. 2020], text [Gal et al. 2022], and sketches [Wang et al. 2021]. Each generative model can represent a substantial investment in a specific idea of the model creator.

As the number of available generative models grows, it is becoming increasingly infeasible for a user to know about every interesting model, and yet it is crucial to choose the right one for their specific use. Each generative model allows a user to easily synthesize an unbounded set of images, interpolations, or latent variable manipulations, but we have found that choosing the right generative model out of a large collection can yield results that are far better than picking a mismatched model (Section 5). Just as information retrieval and image retrieval allow a user to find the right information within vast collections of traditional content, model search enables a user to find a model that best fits their particular needs.

Content-based model retrieval 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 deep 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). A naive Monte Carlo approach can compare the input query to thousands or even millions of samples from each generative model, and identify the model whose samples most often match the input query. Such a sampling-based approach would make model search extremely slow.

To address the above challenges, we first present a general probabilistic formulation of the model search problem and present a Monte Carlo baseline. To reduce the search time and storage, we "compress" the model's distribution into pre-computed 1st and 2nd order moments of the deep feature embeddings of the original samples. We then derive closed-form solutions for model retrieval given an input image, text, sketch, or model query. Our final formula can be evaluated in real-time.

We evaluate our algorithms and perform ablation studies on more than 130 deep generative models such as GANs (e.g., StyleGAN-family models [Karras et al. 2020b]), diffusion models (e.g., DDPM [Ho et al. 2020]), and auto-regressive models (e.g., VQGAN [Esser et al. 2021]). Compared to the Monte-Carlo baseline, our method enables much more efficient search (within 0.08 milliseconds, a 5x speedup), while maintaining high accuracy. Finally, we demonstrate a few applications of model search, including few-shot model fine-tuning [Wang et al. 2018] and GAN inversion [Zhu et al. 2016]. Please view our supplemental video for a model search demo in action. Our search interface, code, and data will be publicly available upon publication.

2 RELATED WORKS

Deep generative models. Generative models are open-sourced at an unprecedented rate of hundreds per month. They use different

learning objectives [Dinh et al. 2017; Goodfellow et al. 2014; Kingma and Welling 2014; Oord et al. 2016; Sohl-Dickstein et al. 2015], training techniques [Karras et al. 2018, 2020a; Kumari et al. 2022; Mokady et al. 2022], 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. 2020a; Mokady et al. 2022; Schultz 2020; Yu et al. 2015] for different applications [Albahar et al. 2021; Ha and Schmidhuber 2018; Lewis et al. 2021; Peebles et al. 2022; Zhang et al. 2021; Zhu et al. 2021a]. This trend leads to the following question. Among all the models, which one shall we use for new tasks and datasets? The goal of our work is *not* to introduce a new model. Instead, we want to help researchers and practitioners find existing models more quickly.

Image editing with generative models. Generative models enable various image editing capacities, thanks to the learned disentangled representations. For example, GAN-based editing methods invert an image to the latent space [Abdal et al. 2020; Bau et al. 2019a; Brock et al. 2017; Roich et al. 2021; Tov et al. 2021; Zhu et al. 2016], and edit the inverted image by modifying the latent code [Härkönen et al. 2020; Ling et al. 2021; Liu et al. 2022; Patashnik et al. 2021; Shen et al. 2020; Zhu et al. 2021b]. Diffusion models [Ho et al. 2020; Song et al. 2021a] transform a user edit (e.g., brushstrokes) to be realistic-looking by adding noise and de-noising [Meng et al. 2022; Ramesh et al. 2022]. Generative transformers [Esser et al. 2021] can be applied to edit images with text-guidance [Crowson et al. 2022] or user-defined masks [Chang et al. 2022]. Besides image editing, several works enable users to edit and customize a generative model. By modifying the network weights directly, we can easily update a pre-trained GAN with simple user interfaces, such as sketching [Wang et al. 2021], blending [Bau et al. 2020], adding/removing certain objects [Bau et al. 2019b], and providing text prompts [Gal et al. 2022]. With the increasing number of generative models, it becomes essential to find the best model to apply image editing. Moreover, the need for a model search system has increased, as a potentially large number of new models can be created by model editing algorithms.

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 to very large collections [Gong et al. 2012; Jégou et al. 2010; Salakhutdinov and Hinton 2009; 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]. When an image is not available, a sketch can serve as a query; sketch-based retrieval has been studied using traditional feature descriptors [Cao et al. 2011; Eitz et al. 2010; Lin et al. 2013; Shrivastava et al. 2011] as well as deep learning methods [Liu et al. 2017; Radenovic et al. 2018; Ribeiro et al. 2020; Sangkloy et al. 2016; Yu et al. 2016]. There has also been 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.

Transfer learning for generative models. Transfer learning aims to adapt models to unseen domains and tasks [Huh et al. 2016; Kornblith et al. 2019; Oquab et al. 2014; Saenko et al. 2010; Yosinski et al. 2014; Zamir et al. 2018]. In the case of generative models like GANs, several works have proposed finetuning a pre-trained generator and discriminator to enable image generation in an unseen limited-data domain [Li et al. 2020; Mo et al. 2020; Noguchi and Harada 2019; Ojha et al. 2021; Wang et al. 2020, 2018; Zhao et al. 2020a]. Various works have explored model selection to choose pre-trained models for discriminative tasks [Bolya et al. 2021; Dwivedi and Roig 2019; Mustafa et al. 2020; Puigcerver et al. 2021] or selecting pre-trained discriminators for training GANs [Kumari et al. 2022]. Similarly, the source of pre-trained generators plays a critical role in GAN finetuning according to a recent study [Ojha et al. 2021]. In our work, we use content-based model search to automatically select pre-trained generators for a new domain, and improve the efficiency of model finetuning.

3 METHODS

We aim to build a search system 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. In this paper we shall focus our attention on unconditional generative models trained on image collections: we are interested in this starting point because the user community has created a growing proliferation of this class of models [Pinkney 2020a]. We denote the model collection by $\theta \sim \text{unif}\{\theta_1, \theta_2, \ldots, \theta_N\}$ and the user query by q. Every model θ_n captures a distribution of images $p(x|\theta)$. Since prior retrieval methods [Manning et al. 2010; Smeulders et al. 2000] search for single instances, our key challenge is to establish the notion of retrieving probabilistic distributions.

To achieve this, we introduce a probabilistic formulation for generative model retrieval. 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, and we present our model retrieval algorithms for an image, a text, and a sketch query, respectively. In Section 3.2, we demonstrate several extensions and applications of our search system, including finding similar models, editing real images, and fine-tuning GANs with a few images. In Section 3.3, we present a user interface built upon our algorithms.

3.1 Probabilistic Retrieval for Generative Models

Our goal is to quantify the likelihood of each model θ_n given the user query q, by evaluating the conditional probability $p(\theta|q)$. The model with the highest conditional probability is retrieved:

$$\max_{\theta \in \{\theta_1, \ldots, \theta_N\}} p(\theta|q). \tag{1}$$

It is difficult to assess $p(\theta|q)$ directly, since there is no known paired query-to-model datasets. We address this problem by modeling the joint distribution of models and images conditional on the query:

$$\begin{aligned} p(\theta, x|q) &= p(\theta|x, q)p(x|q) \\ &= p(\theta|x)p(x|q), \end{aligned} \tag{2}$$

where we assume the model θ and query q are conditionally independent given the image x. $p(\theta|q)$ is then the integral of the joint distribution over x:

$$\begin{aligned} p(\theta|q) &= \int p(\theta, x|q)dx \\ &= \int p(\theta|x)p(x|q)dx \\ &= p(\theta)\int \frac{p(x|\theta)}{p(x)}p(x|q)dx. \end{aligned} \tag{3}$$

Finally, since we assume θ is uniformly distributed, we can omit the prior term $p(\theta)$ and solve for:

$$\max_{\theta\in\{\theta_1,\ldots,\theta_N\}} \int \frac{p(x|\theta)}{p(x)}p(x|q)dx. \tag{4}$$

Our formulation reduces model retrieval into two well-studied problems. We can (1) first estimate $p(x|\theta)$ from the density of the generative model θ and (2) then compute $p(x|q)$ based on image similarity or cross-modal similarity. Unfortunately, the integral over x remains intractable. To resolve this issue, we approximate the integral for image and text queries in the following.

Image-based model retrieval. Given an image query, the best matched image will be itself. Hence, we model $p(x|q)$ as a Dirac delta function $\delta(x-q)$, and we can reduce the problem as follows.

$$\max_{\theta\in\{\theta_1,\ldots,\theta_N\}} \frac{p(q|\theta)}{p(q)} \propto p(q|\theta) \tag{5}$$

Equation 5 indicates that the best matched model is the most likely one to generate the query image. Since the density is intractable for implicit generative model (e.g., GANs [Goodfellow et al. 2014]), we approximate each model by a Gaussian distribution of image features [Heusel et al. 2017a]. For an image x, we obtain image features $z := \psi_{\text{im}}(x)$, where ψ_{im} is the feature extractor. Now we express Equation 5 in terms of image features z.

$$\max_{n\in\{1,\ldots,N\}} (2\pi|\Sigma_n|)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(z_q-\mu_n)^T\Sigma_n^{-1}(z_q-\mu_n)\right) \tag{6}$$

where the query image feature is denoted by $z_q := \psi_{\text{im}}(q)$, and each model θ_n is approximated by $p(z|\theta_n) \sim \mathcal{N}(\mu_n, \Sigma_n)$. We refer to this method as `Gaussian Density`. In Section 4, we show that our method can retrieve models that share similar visual concepts with the query. Moreover, we can directly apply our image-based model retrieval method on sketches.

Sketch-based model retrieval. We can use the same method for sketch-based model retrieval if the embedding network ψ_{im} also works for human sketches. In our experiment, we find that CLIP [Radford et al. 2021] can produce similar feature embeddings for similar image and sketches. CLIP outperforms other pre-trained networks (e.g., DINO [Caron et al. 2021]) by a large margin.

Text-based model retrieval. A text query may correspond to multiple possible image matches $p(x|q)$, so that we cannot assume $p(x|q)$ to be a Dirac delta function as before.

Instead, we estimate the term $\frac{p(x|q)}{p(x)}$ in Eqn. 4. We note that this expression is proportional to the score function f in contrastive learning (e.g., InfoNCE [Oord et al. 2018]), where $f(x,q) \propto \frac{p(x|q)}{p(x)}$. In fact, since CLIP [Radford et al. 2021] is trained on a text-image retrieval task with the InfoNCE loss, we can directly apply pre-trained CLIP model to simplify Eqn. 4.

$$\max_{\theta\in\{\theta_1,\ldots,\theta_N\}} \int p(x|\theta)f(x,q)dx \propto \mathbb{E}_{x\sim p(x|\theta)}\left[f(x,q)\right]. \tag{7}$$

We recall that CLIP consists of an image encoder ϕ_{im} and a text encoder ϕ_{txt}, and it is trained with a score function based on cosine similarity:

$$f(x,q) = \exp\left(\frac{{h_x}^T h_q}{\tau(||h_x|| \cdot ||h_q||)}\right) = \exp\left(\frac{\tilde{h_x}^T \tilde{h_q}}{\tau}\right), \tag{8}$$

where $h_x := \phi_{\text{im}}(x)$ and $h_q := \phi_{\text{txt}}(q)$ are the image and text feature from CLIP, respectively. $\tilde{h_x} = \frac{h_x}{||h_x||}$ and $\tilde{h_q} = \frac{h_q}{||h_q||}$ are the normalized features. Hence, Eqn. 9 can be written precisely as:

$$\max_{\theta\in\{\theta_1,\ldots,\theta_N\}} \mathbb{E}_{h_x\sim p(h_x|\theta)}\left[\exp\left(\frac{\tilde{h_x}^T \tilde{h_q}}{\tau}\right)\right] \tag{9}$$

Now we have a tractable Monte Carlo estimate of the integral. We sample images from each model and average the score function of each image sample x and the text query q. We refer to this method as `Monte-Carlo`. However, directly applying Monte Carlo estimation is not efficient in practice, since we need lots of samples to yield a robust estimate. To speed up computation, we provide two ways to approximate Eqn. 9. First, we find that a point estimate at the first moment of $p(h_x|\theta)$ works well. We directly estimate the cosine distance between the mean image features and the query feature. Since the exponential and temperature mapping is monotonically increasing, the matching function becomes:

$$\begin{gathered} \max_{\theta\in\{\theta_1,\ldots,\theta_N\}} \tilde{\mu_n}^T \tilde{h_q}, \\ \text{where } \mu_n = \left(\mathbb{E}_{h_x\sim p(h_x|\theta_n)}\left[h\right]\right);\ \tilde{\mu_n} = \frac{\mu_n}{||\mu_n||} \end{gathered} \tag{10}$$

We refer to this method as 1^{st} `Moment`. We can also approximate $p(h_x|\theta)$ using both the first and the second moment to get the following expression.

$$\begin{gathered} \max_{\theta\in\{\theta_1,\ldots,\theta_N\}} \frac{1}{2\tau}\tilde{h_q}^T\Sigma_n\tilde{h_q} + u_n^T\tilde{h_q} \\ \text{where } p(h_x|\theta_n) \sim \mathcal{N}(\mu_n, \Sigma_n) \text{ for } n \in \{1,\ldots,N\}, \end{gathered} \tag{11}$$

We refer to this method as 1^{st} + 2^{nd} `Moment`. We provide the details of the derivation in the supplement. Empirically, the performance is similar between approximation to the first or second moment. We provide more analysis in Section 4.

3.2 Extensions and Applications

Here we show additional extensions of our system. Our model search system can also facilitate several computer graphics and vision applications.

Multi-modal query. We can further extend our model search to handle multiple queries from different modalities. To achieve this, we use a Product-of-Experts formulation [Hinton 2002; Huang et al. 2021] wherein the final likelihood of a model, given a multimodal query (e.g., text-image pair), is modeled as a product of likelihoods given individual queries followed by a renormalization.

Finding similar models. Once a model is found, we enable navigation to similar models. To compute similarity between models, we use the Fréchet Distance [Dowson and Landau 1982] between the models' feature distributions. Following prior work [Heusel et al. 2017a; Kynkäänniemi et al. 2022], we approximate a model's distribution by fitting a multivariate Gaussian in an image feature space. Then the Fréchet Distance can be computed directly from the Gaussian parameters. For each model, we pre-compute a list of similar models based on the smallest pairwise distances.

Real image editing. There are many GAN-based image editing methods [Brock et al. 2017; Ling et al. 2021; Patashnik et al. 2021; Tov et al. 2021; Zhu et al. 2016] that create realistic changes in an image by manipulating the latent variables of a pre-trained generator. However, these methods all assume that we begin with a generative model that matches the image's domain. In the wild, a user may have an image without a corresponding generative model. Given a collection of generative modes, our system automatically finds a suitable model to perform image edits. Starting with a input real image, we apply our image-based model retrieval method (Section 3.1) to find the best matched model. We find that the best matched model is suitable for inversion and image editing (Table 4, Figure 8, Figure 9).

Few-shot fine-tuning. Fine-tuning from a pre-trained model is one of the standard methods to train generative models on a limited amount of data [Karras et al. 2020a; Wang et al. 2018; Zhao et al. 2020b]. It helps in mitigating overfitting and requires less compute resources as well. The abundance of pre-trained generative models—which inevitably will further increase—presents a unique problem of finding the best base model to fine-tune on a small number of images from a new domain. In Section 5.4, we show empirically that transfer learning from similar generative models as selected by our retrieval methodology leads to on average faster convergence and better performance in a new domain with limited data.

3.3 User Interface

We create a web-based UI for our search algorithm. The UI supports searching and sampling from deep generative models in real time. The user can enter a text prompt, upload an image/sketch, or provide both text and an image/sketch. The interface displays the models that match most closely with the query. Clicking a model takes the user to a new page where they can sample new images from the model. The website employs a backend GPU server to enable real time model search and image synthesis capabilities even on a mobile device. Figure 3 shows a screenshot of our UI. For more details, please watch the accompanying video demo.

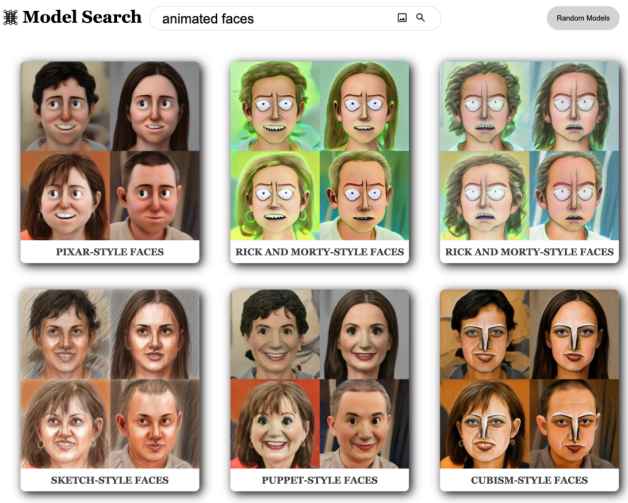

Fig. 3. **User interface of model search.** The user can enter a text query and/or upload an image in the search bar to retrieve generative models that best match the query. Here we show top retrievals for the text query "animated faces," which shows StyleGAN-NADA models [Gal et al. 2022] trained on animated characters. The user can further explore randomly generated images from the models or search for more similar models to a particular model. Please view the supplemental video for more details.

4 EXPERIMENTS

Here we first evaluate our model retrieval methodology over text, image, and sketch modalities and discuss several algorithmic design choices. We then show qualitative and quantitative results for the extensions and applications enabled by our model search.

Generative model zoo. We evaluate on a collection of 133 generative models trained using different techniques including GANs [Gal et al. 2022; Karras et al. 2018, 2021, 2020b; Kumari et al. 2022; lucid layers 2022; Mokady et al. 2022; Pinkney 2020a,b; Sauer et al. 2022; Wang et al. 2021], diffusion models [Dhariwal and Nichol 2021; Ho et al. 2020; Song et al. 2021b], MLP-based generative model CIPS [Anokhin et al. 2021], and the autoregressive model VQGAN [Esser et al. 2021]. We also assign tags to each model based on the type of generated images, with 23 tags in total. Example tags include "face", "animals", "indoor", where all the face generative models will have the "face" tag. Similarly, models trained on datasets like bedroom and conference categories of LSUN [Yu et al. 2015] are tagged as "indoor".

Implementation details. For image-based model retrieval and similar model search, we test on three different image features – Inception [Szegedy et al. 2016], CLIP [Radford et al. 2021], and DINO [Caron et al. 2021]. For text-based model retrieval, we use CLIP features, as discussed in Section 3.1. We note that CLIP learns a magnitude-invariant feature space, since it is trained by maximizing cosine similarities between text and image features. Hence, we choose to use ℓ_2-normalized CLIP features, which also outperforms its unnormalized version empirically.

To efficiently evaluate our method and compare it across different baselines, we pre-compute and save 50K generated image features

for each model in the CLIP, DINO, and Inception feature space. Similarly, for the `Gaussian Density` based method, we pre-calculate and save the mean and covariance. To calculate features, we follow the pre-processing steps as proposed by Parmar et al. [2021].

4.1 Model Retrieval

Evaluation metrics. Following the image retrieval literature [Philbin et al. 2007; Weyand et al. 2020], we evaluate model retrieval using two different metrics: (1) Top-k accuracy, i.e., predicting the ground truth generative model of each query in top k, and (2) Mean Average Precision@k (mAP@k) [Weyand et al. 2020]. All the models with tags common to the query model are included as relevant models regarding the mAP calculation. The mAP@k metric considers only the top-k predictions as for retrieval use-cases, top-ranked models matter more. This metric is computed as below:

$$mAP@k = \frac{1}{N}\sum_{i=1}^{N} AP@k(q_i),$$
$$\text{where } AP@k(q) = \frac{1}{\min(GT_q, k)}\sum_{j=1}^{k} P_q(j)Rel_q(j), \tag{12}$$

where $P_q(j)$ is the precision of top-j predictions given query q and $Rel_q(j)$ is a binary indicator for j^{th} prediction being relevant. GT_q is the number of relevant models corresponding to the query. The above metric weighs all the models similar to the query ground truth equally during evaluation. For example, given a text query *face*, all face generative models are treated as relevant and should be retrieved as top predictions.

Text-based model retrieval. To evaluate text retrieval, we manually assign a ground-truth text description to each model in the collection. We use CLIP feature space [Radford et al. 2021], since CLIP has both text and image encoders. In Table 1, we show retrieval performance of the different methods as discussed in Section 3. We achieve top-10 accuracy of 92% and 0.71 score of mAP@10 applying the `1st Moment` based method. Compared to the `Monte-Carlo` approach with 50K samples, it is more than 5 times faster while performing similarly (as shown in Table 3). To ensure that the retrieval is robust to variation in text queries, we also evaluate the method with augmented queries that prepend phrases like "an image of" and "a photo of" to each text query. As shown in Table 1, the retrieval performance decreases only marginally. Figure 4 shows qualitative examples of the top three and lowest ranked retrieval given a text query. Both quantitative numbers and visual inspection of results show that our method retrieves relevant generative models.

Image- and sketch-based model retrieval. To evaluate image-based model retrieval, we automatically create image queries using images generated by each model. We generate 50 image queries for each model and use the corresponding model as the ground truth for the queries. In total we have 133×50 image queries. To obtain sketch queries, we use the method of Chan et al. [2022] and PhotoSketch [Li et al. 2019] to convert images into sketches.

We also apply `1st Moment` method (Eqn. 10) to image-based model retrieval. Specifically, given an image query, we extract the feature using CLIP's image encoder ϕ_{im}. We then compute the cosine distance between the query feature and the first moment μ_n.

		Top-k Accuracy			mAP@k		
		Top-1	Top-5	Top-10	mAP@5	mAP@10	mAP
Original	Monte-Carlo (50K)	0.49	0.78	0.92	0.76	0.71	0.71
	Monte-Carlo (1K)	0.49	0.78	0.92	0.75	0.71	**0.71**
	1st Moment (ours)	**0.50**	**0.80**	**0.92**	**0.78**	**0.71**	0.68
	1st + 2nd Moment (ours)	0.49	0.79	0.92	0.74	0.70	0.70
Augmented	Monte-Carlo (50K)	0.49	0.80	0.92	0.79	0.73	0.73
	Monte-Carlo (1K)	0.50	0.79	0.92	0.78	**0.73**	**0.72**
	1st Moment (ours)	**0.51**	**0.80**	**0.94**	**0.79**	0.72	0.69
	1st + 2nd Moment (ours)	0.50	0.78	0.92	0.77	0.72	0.72

Table 1. **Text-based model retrieval**. Our `1st Moment` and `1st + 2nd Moment` methods perform on-average similar to the Monte-Carlo based approach while being more computationally efficient. We evaluate the retrieval using top-k and mAP@k evaluation metrics. For the Augmented version, we create multiple queries from each text query by pre-pending phrases like "an image of" in front of the query.

		Top-k Accuracy			mAP@k		
		Top-1	Top-5	Top-10	mAP@5	mAP@10	mAP
Image (Gen.)	CLIP+Monte-Carlo (50K)	0.82	**0.98**	1.00	0.81	0.74	0.75
	CLIP+1st Moment (ours)	0.75	0.95	0.99	0.79	0.74	0.74
	CLIP+Gaussian Density (ours)	0.77	0.95	**1.00**	**0.81**	**0.75**	**0.76**
	DINO+Gaussian Density	**0.83**	0.96	0.98	0.80	0.73	0.69
	Inception+Gaussian Density	0.70	0.92	0.98	0.79	0.72	0.67
Chan et al. [2022]	CLIP+Monte-Carlo (50K)	**0.36**	0.69	0.86	0.67	0.63	0.65
	CLIP+1st Moment (ours)	0.35	**0.70**	0.84	0.70	0.64	0.64
	CLIP+Gaussian Density (ours)	0.33	0.67	**0.86**	**0.68**	**0.64**	**0.66**
	DINO+Gaussian Density	0.08	0.24	0.32	0.32	0.26	0.33
	Inception+Gaussian Density	0.08	0.22	0.29	0.30	0.25	0.34
Photo-sketch	CLIP+Monte-Carlo (50K)	0.11	0.30	0.44	0.36	0.34	0.42
	CLIP+1st Moment (ours)	**0.11**	0.30	0.45	**0.48**	**0.43**	**0.46**
	CLIP+Gaussian Density (ours)	0.11	**0.31**	**0.47**	0.41	0.39	0.46
	DINO+Gaussian Density	0.01	0.06	0.09	0.06	0.04	0.19
	Inception+Gaussian Density	0.01	0.07	0.12	0.10	0.09	0.21

Table 2. **Image- and sketch-based model retrieval**. We evaluate retrieval in the feature space of CLIP, DINO, and Inception networks and observe best performance using CLIP. We use the generated images as query for image-based model retrieval. For sketch-based evaluation we use the method of Chan et al. [2022] and Photo-sketch [Li et al. 2019] to convert generated images to sketch.

In Table 2, we show retrieval results of different formulations with CLIP, DINO, and Inception network features. We observe that the retrieval performance is best with CLIP features across different query types, especially for sketches. For image-based retrieval, `Gaussian Density` outperforms `Monte-Carlo` and `1st Moment` on the mAP metric. In terms of speed, `1st Moment` method performs the best ($\sim$ 5 times faster) at the cost of worse performance, compard to `Gaussian Density`. For sketch-based retrieval, CLIP features significantly outperform DINO and Inception features. Example retrieval results are shown in Figure 4 for both image and sketch queries. Figure 5 further shows qualitative ablation of the three pretrained networks' feature space. For a dog sketch query, both DINO and Inception features fail to retrieve the relevant model and instead return art-based models.

Running time and memory. The computational and memory efficiency of the retrieval algorithm is crucial to supporting many concurrent users searching over large-scale model collections. Therefore. we profile our method on 133 generative models as well as a

Fig. 4. **Qualitative results of model retrieval.** *Top row (image query):* The still-life painting retrieves models related to art and places the AFHQ Wild [Choi et al. 2020b] model at the bottom ranking. *Middle row (sketch query)* Both horse and church sketches retrieve relevant models at the top ranking. *Bottom row (text query):* The query "human wearing a pair of glasses" successfully retrieves a GANSketch [Wang et al. 2021] model finetuned for human faces with glasses. Similarly, with the query "a bird that talks", we find a Self-Distilled GAN [Mokady et al. 2022] trained on Internet parrots images.

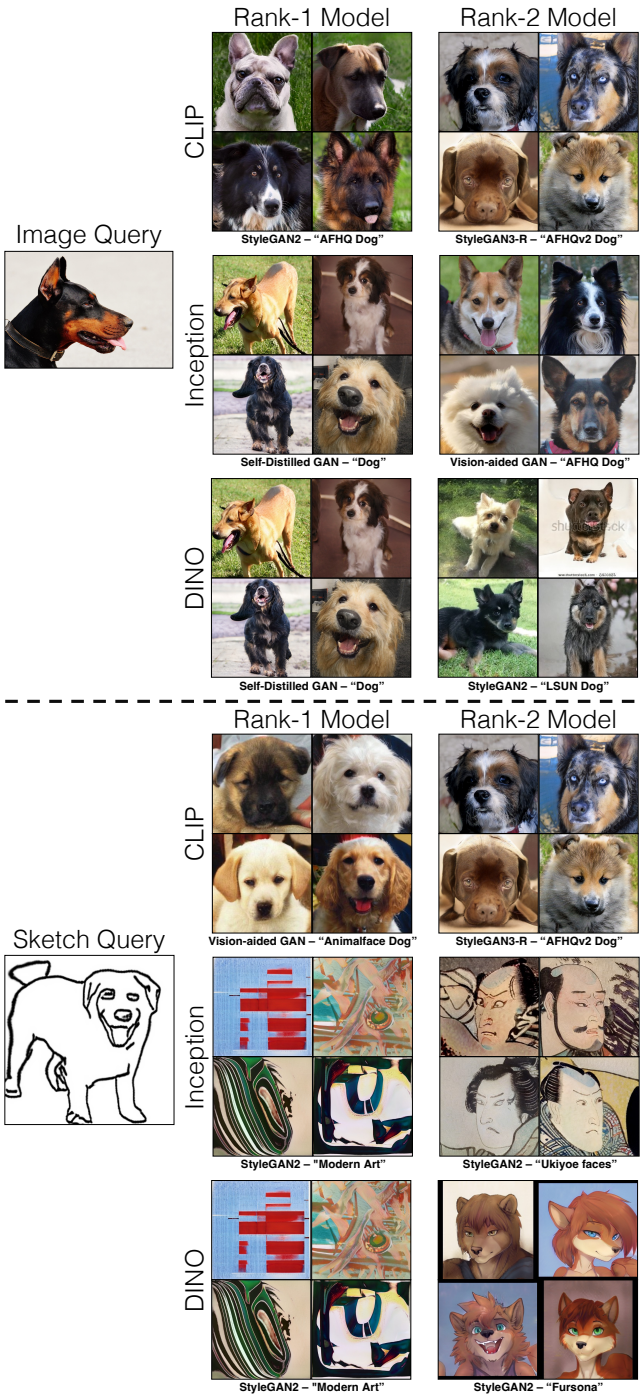

Fig. 5. **Qualitative comparison of image- and sketch-based model retrieval in different networks' feature spaces.** For image queries, all three networks—CLIP, DINO, and Inception have similar performance and return relevant models. For sketch queries, we observe that CLIP works significantly better, as shown in the example above and in Table 2. Both inception and DINO score artistic models higher, which might not be best suited for the given query.

		Feature Extraction	Model Scoring		
			133	10K	1M
Text	Monte-Carlo (50K)	5.02ms	0.46ms	OOM	OOM
	1^{st} Moment		**0.08ms**	**0.20ms**	**0.29ms**
Image	Monte-Carlo (50K)	6.75ms	0.46ms	OOM	OOM
	Gaussian Density		0.44ms	0.46ms	OOM
	1^{st} Moment		**0.08ms**	**0.20ms**	**0.29ms**

Table 3. **Model retrieval running time.** Both the 1st Moment and Gaussian Density-based scoring methods run quickly on our test machine. After we get the score for each model, we additionally run `torch.argsort` to get the best matches, which takes 0.04, 0.11, and 0.07ms for 133, 10K, and one million models, respectively (not shown in the table). OOM stands for "out of memory" and that the retrieval cannot be done in a single pass.

much greater number of simulated models. To create simulated models, we sample the 1^{st} `Moment` model statistics and `Monte-Carlo` samples from a 512-dimensional normal distribution that correspond to points in the CLIP feature space, and the second moment statistics are generated as a unit covariance matrix with a small, uniform, symmetric noise added. The data are stored in the GPU's VRAM. We run the following tests on a machine equipped with an AMD Threadripper 3960X and NVIDIA RTX A5000 running Pytorch 1.11.0, and report them in Table 3.

For text-based model retrieval, extracting query features with CLIP takes 4.95 ms and 0.36 GB of VRAM (this does not scale with the number of models). For 133 models, the full 50K-sample Monte-Carlo method takes 0.46 ms but uses 12.87 GB of VRAM, making memory a constraint for retrieving many models in a single pass. On the other hand, the 1^{st} `Moment` based method is much more computationally efficient and almost as precise (see Table 2). For one million models, the 1^{st} Moment based scoring method takes 0.28 ms and 1.93 GB of VRAM. When handling image-to-model or sketch-to-model retrieval, extracting query features with CLIP takes 7.40 ms and 0.36 GB of VRAM. When 1^{st} + 2^{nd} `Moment` is used to score models for visual queries, extra memory is needed to store the models' covariance matrices. With 10,000 models, it takes 0.44ms and 9.85 GB of VRAM. Switching to the 1^{st} `Moment` only method improves the computational efficiency, enabling sketch- and image-based retrieval from one million models in a single pass, albeit at a slight reduction in precision (see Table 2).

After scoring is done, sorting the scores and obtaining the most relevant models runs quickly on a GPU. For one million models, sorting takes 0.12ms and 0.062 GB of VRAM. Therefore, our 1^{st} `Moment` method, a user can retrieve models from a 1-million-model collection with a text, sketch, or image query in real-time.

5 EXTENSIONS AND APPLICATIONS

Our work enables users to explore available generative models and find the best models for different use cases. Here we show several use cases, including multi-modal queries, finding similar models, image editing, and few-shot transfer learning. For all applications we used CLIP feature space in our retrieval method as it performs the best across modalities.

Fig. 6. **Multi-modal user query.** We show a qualitative example of how multi-modal queries can help refine the model search. With only the image of "Nicolas Cage" we retrieve only face models. But with the multi-model query of image and text as "dog", we can retrieve the StyleGAN-NADA model of "Nicolas Cage dogs".

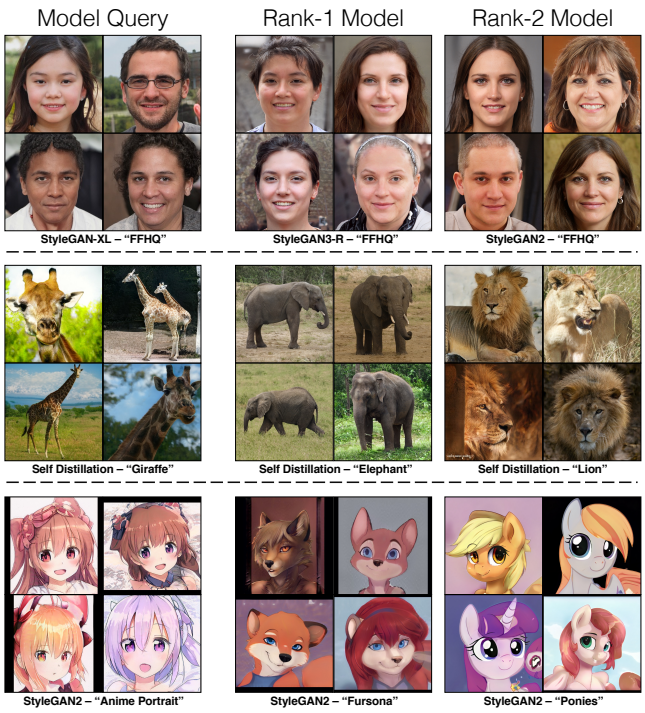

Fig. 7. **Finding similar models.** In the top row, when the query is a face model, we retrieve more face generative models. In the middle and bottom rows, in which we our collection does not contain models in the identical category, our method returns models of broadly similar categories.

5.1 Multimodal User Query

We show qualitatively that our search method can be extended to multimodal queries, based on the Product-of-Experts formulation described in Section 3.2. We demonstrate how leveraging multiple input modalities from the user can retrieve models which are better tailored to user queries. Specifically, we test this application for text-image pairs, as shown in Figure 6.

5.2 Finding Similar Models

As explained in Section 3.2, we use the FID between the feature distribution of each generative model as the scoring method for retrieving similar models. We use CLIP, DINO, and Inception networks' feature space and evaluate Average Precision using ground truth similar models. We get an AP of 0.68, 0.68, and 0.66 respectively. Figure 7 shows qualitative examples of similar model retrieval using FID metric in CLIP feature space.

5.3 Image Reconstruction and Editing

Image inversion. For image editing, we use 15 StyleGAN2-based models at 256x256 resolution, for a fair comparison regarding image resolution and network architectures. We first evaluate real image inversion on validation set images from LSUN Church [Yu et al. 2015] and CelebA-HQ datasets [Karras et al. 2018]. We use the optimization-based inversion technique [Karras et al. 2020b] in W^+ latent space with LPIPS [Zhang et al. 2018] and pixel level mean square loss between the target and generated image. Given an image query, we run inversion on models ranked at 1, 10, and 15 by our `Gaussian Density` retrieval method. We evaluate the reconstruction quality between 100 inverted and target images of both category using standard metrics like LPIPS and PSNR. We also calculate the mean distance between optimized w^+_{opt} and the mean latent w_{avg} of the model which shows the extent of overfitting to the reconstruction loss by the model. The results are as shown in Table 4. We observe that top retrieved models that are similar to the image query result in better image inversion on-average across all metrics. Moreover, for lower ranked models, the distance $||w^+_{opt} - w_{avg}||$ is significantly higher compared to the top-rank models, which has been shown to correlate negatively with image editability [Tov et al. 2021; Zhu et al. 2020]. Figure 8 shows some qualitative samples of image inversion using the different ranked models. Lower-ranked models yield inversions with substantially poorer quality.

Image editing and interpolation. We now show that images inverted with top-ranked models can be further edited using existing GAN-based image editing techniques such as GANSpace [Härkönen et al. 2020]. Figure 9 shows examples of editing on Ukiyo-e images to change the frowning face to a smiling face. For the flower category, we show example edits that change the petal colors. We can also perform latent space interpolation between inverted images of the same category and create visually compelling samples as shown in Figure 11. A rank-1 model results in smoother interpolation of the

Dataset	Image Inversion											
	LPIPS-alex (↓)			LPIPS-vgg (↓)			PSNR(↑)			$\|\|w^+_{opt} - w_{avg}\|\|$ (↓)		
	Rank-1	Rank-10	Rank-15	Rank-1	Rank-10	Rank-15	Rank-1	Rank-10	Rank-15	Rank-1	Rank-10	Rank-15
CelebA-HQ	**0.13**	0.31	0.28	**0.21**	0.37	0.31	**23.94**	20.68	22.20	**803.18**	1164.52	2291.42
LSUN Church	**0.32**	0.38	0.39	**0.36**	0.43	0.44	17.91	17.78	**18.39**	**570.90**	1194.89	1759.43

Table 4. **Inverting real images using different ranked models**. We use 100 images each of CelebaHQ and LSUN Church dataset as queries and invert each image using models at 1, 10 and 15 rank in the retrieval score. The reconstruction quality is measured using LPIPS [Zhang et al. 2018] and PSNR. For both datasets, top-ranked models are significantly better at image inversion compared to lower-rank models. We also measure the mean distance between the optimized latent w^+_{opt} and mean latent w_{avg} of the model. For lower-rank models, the optimized latent w^+_{opt} deviates considerably from the mean latent which hints at overfitting over reconstruction loss and low image editability [Tov et al. 2021; Zhu et al. 2020].

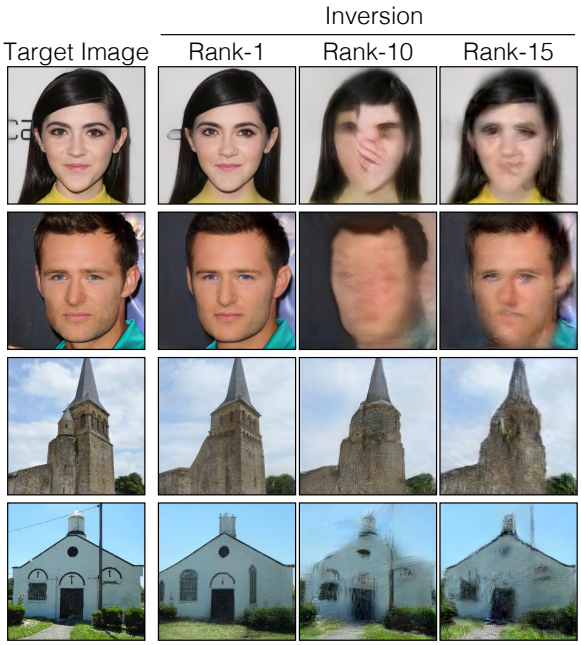

Fig. 8. **Projecting real images to retrieved StyleGAN2 models.** Example image inversion for CelebA-HQ and LSUN Church images using different ranked models. Given the query image (1st column), using the top-ranked model results in more accurate image reconstruction compared to lower-ranked models. Models at rank-1, 10, and 15 that are retrieved for the above queries are as follows. *Row 1 and 2:* StyleGAN2-FFHQ, StyleGAN2-LSUN Church, Vision-aided StyleGAN2-100-shot-Bridge. *Row 3:* Vision-aided StyleGAN2-LSUN Church, StyleGAN2-Cakes, Vision-aided StyleGAN2-Animalface Cat. *Row 4:* Vision-aided StyleGAN2-LSUN Church, StyleGAN2-100-shot-Bridge, Vision-aided StyleGAN2-Animalface Cat.

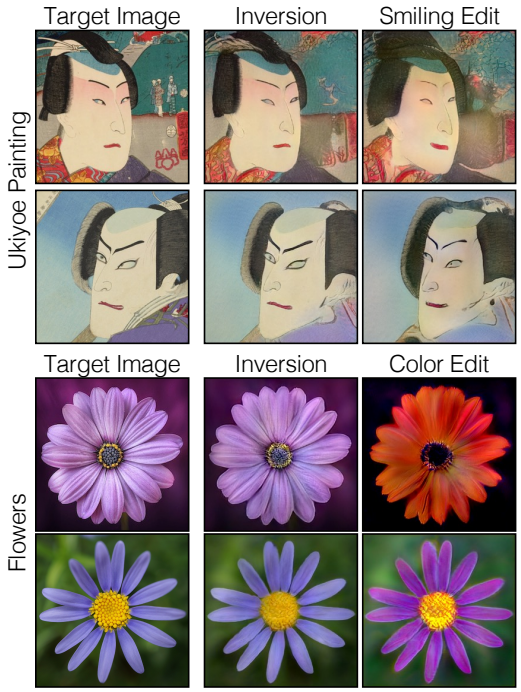

Fig. 9. **Editing real images.** We use the model ranked first by our image-based model retrieval algorithm for inverting the real image, and then we perform editing using GANspace [Härkönen et al. 2020]. Since using a random model often leads to sub-optimal inversion, selecting relevant models is critical for image editing applications.

one image into another in contrast to inversion and interpolation using a random model.

5.4 Few-Shot Transfer Learning

For few-shot transfer learning, we restrict our experiments to 256x256 resolution StyleGAN2 models due to limited computing resources (27 models). We begin with a small dataset of 100-136 images. Then, we rank the 27 models by the average retrieval score over all images. We select models at rank 1, 20 and 25 as source models for transfer learning. For finetuning the generator on the new dataset, we use vision-aided GANs [Kumari et al. 2022], one of the leading methods in few-shot GAN training.

Datasets and evaluation metric. We use three standard few-shot datasets: Obama [Zhao et al. 2020b] (100 images), Grumpy Cat [Zhao et al. 2020b] (100 images), and Moongate [Liu et al. 2021] (136 images). We use the Fréchet Inception Distance (FID) [Heusel et al. 2017b] metric for evaluation.

Results. Figure 10 shows the results of transfer learning using different source models with varying retrieval rankings. We observe on average faster convergence when finetuning from rank-1 models compared to other lower-ranked models. This shows empirically that training from similar models results in faster convergence and thus requires less compute.

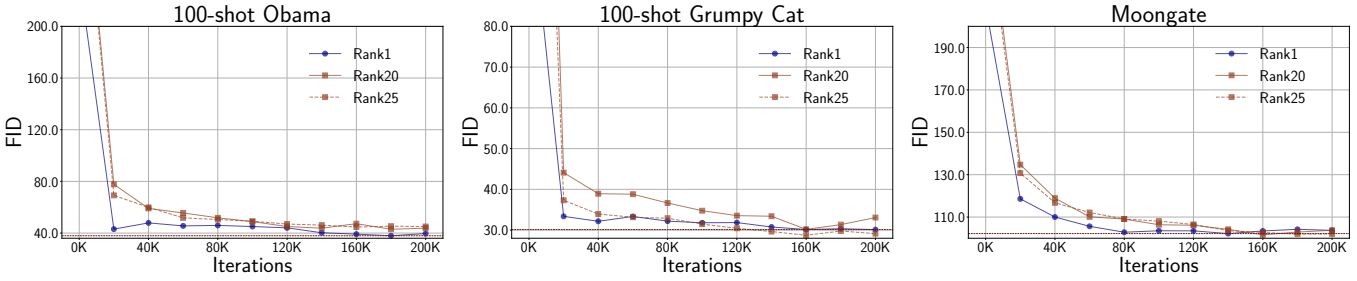

Fig. 10. **Few-shot transfer learning with model search.** Using generative models similar to the source dataset as ranked by our model retrieval algorithm leads to faster convergence and better or on-par performance in terms of FID metric. *Left*: 100-shot Obama dataset finetuned using rank-1 StyleGAN2-FFHQ model and rank-20 and rank-25 models trained on LSUN Horse and Church respectively. *Middle*: 100-shot Grumpy cat dataset finetuned using rank-1 Vision-aided-Animalface Cat model and rank-20, 25 models trained on cakes and LSUN Horse respectively. *Right*: Moongate dataset finetuned using rank-1 StyleGAN2 LSUN Church model and rank-20 and rank-25 models trained on LSUN Cat and Horse respectively.

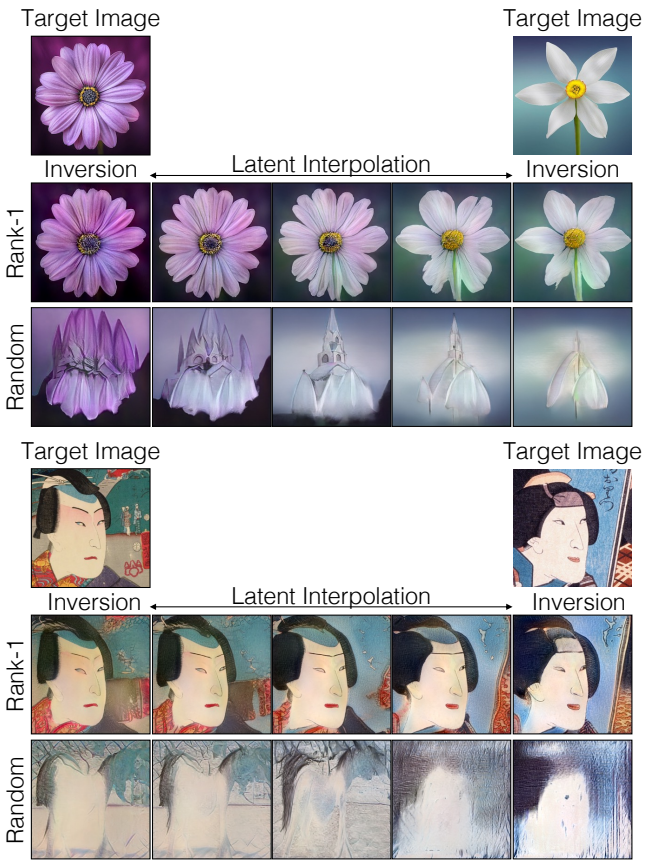

Fig. 11. **Latent space interpolation using rank-1 models vs random model.** Our model search algorithm can be used to create new image interpolations by first finding the most relevant model for inverting images and then interpolating in that generative model's latent space. Selecting a relevant model leads to meaningful interpolation between the two images.

6 DISCUSSION AND LIMITATIONS

We have introduced the problem of content-based retrieval for deep generative image models, whose goal is to help users find, explore,

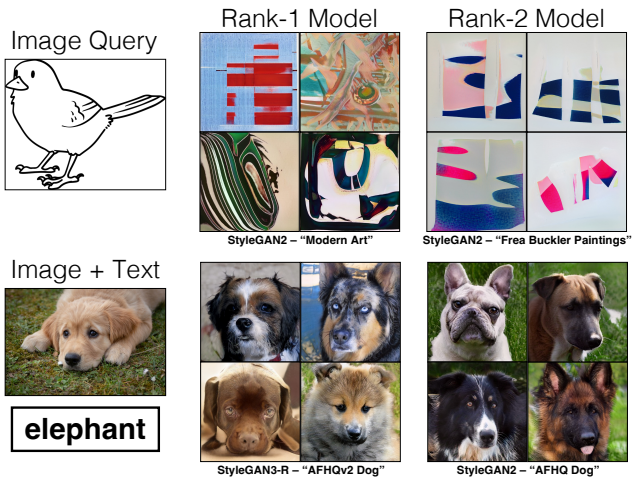

Fig. 12. **Failure cases.** (Top) Sometimes, a sketch query (e.g., the bird sketch) will match models with abstract styles. It is ambiguous whether the CLIP feature should match the shape of the sketch, or the styles and textures. (Bottom) For conflicting multi-modal queries (`elephant` text query + a dog image), our system has difficulty retrieving models with both concepts. There are no elephant models in the top-ranked models.

and share new generative models more easily. Interestingly, we have found that scoring based on a probabilistic model works well, and further that applying a Gaussian density or first-moment approximation to the distribution of generated image features produce accurate search results with a minimal memory and time footprint. We have demonstrated a model search prototype using our method. Our experiments have shown that searches over an indexed collection are useful for finding a good model for image editing and transfer learning.

In Figure 12, we show several limitations of our current method: queries for specific sketches (e.g., the bird sketch in the figure) will sometimes match models that generate a wide range of abstract shapes rather than the specific intended model. Conversely, queries intended to capture diversity will sometimes match an overly specific model. Developing new ways to allow a user to describe the desired diversity in a model is a promising area for future work.

Further, our method is susceptible to failure when the query itself is inherently ambiguous. For instance, given a sketch input of a cat, our method cannot recognize if the user intends to retrieve a model that generates cat sketches or a model that generates real looking cat images conditioned on the sketch. Finally, it is difficult for our method to handle conflicting multimodal queries (e.g., `elephant` text query + a dog image).

Our study is a small first step. We have demonstrated a way to search over unconditional generative models trained on image datasets. Still, we have not yet examined conditional models nor models that synthesize text, audio, or other media. Nevertheless, as collections of many kinds of pretrained models continue to balloon, we have shown that model search is a feasible approach for working with model collections, and we anticipate that, coupled with effective search methods, collections of pretrained models will be an increasingly valuable resource for practitioners and researchers.

REFERENCES

Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. Image2StyleGAN++: How to Edit the Embedded Images?. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

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 Transactions on Graphics (TOG)* (2021).

Ivan Anokhin, Kirill Demochkin, Taras Khakhulin, Gleb Sterkin, Victor Lempitsky, and Denis Korzhenkov. 2021. Image Generators with Conditionally-Independent Pixel Synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Relja Arandjelović and Andrew Zisserman. 2012. Three things everyone should know to improve object retrieval. In *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2911–2918.

Derek Philip Au. 2019. This vessel does not exist. https://thisvesseldoesnotexist.com/.

Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural codes for image retrieval. In *European conference on computer vision*. Springer, 584–599.

Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. 1999. *Modern information retrieval*. Vol. 463. ACM press New York.

David Bau, Steven Liu, Tongzhou Wang, Jun-Yan Zhu, and Antonio Torralba. 2020. Rewriting a deep generative model. In *European Conference on Computer Vision (ECCV)*.

David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. 2019a. Semantic Photo Manipulation with a Generative Image Prior. *ACM Transactions on Graphics (Proceedings of ACM SIGGRAPH)* 38, 4 (2019).

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

Amit H Bermano, Rinon Gal, Yuval Alaluf, Ron Mokady, Yotam Nitzan, Omer Tov, Or Patashnik, and Daniel Cohen-Or. 2022. State-of-the-Art in the Architecture, Methods and Applications of StyleGAN. *arXiv preprint arXiv:2202.14020* (2022).

Daniel Bolya, Rohit Mittapalli, and Judy Hoffman. 2021. Scalable Diverse Model Selection for Accessible Transfer Learning. *Advances in Neural Information Processing Systems* 34 (2021).

Andrew Brock, Jeff Donahue, and Karen Simonyan. 2019. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations (ICLR)*.

Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. 2017. Neural photo editing with introspective adversarial networks. In *International Conference on Learning Representations (ICLR)*.

Yang Cao, Changhu Wang, Liqing Zhang, and Lei Zhang. 2011. Edgel index for large-scale sketch-based image search. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *IEEE International Conference on Computer Vision (ICCV)*. 9650–9660.

Caroline Chan, Fredo Durand, and Phillip Isola. 2022. Learning to generate line drawings that convey geometry and semantics. *arXiv preprint arXiv:2203.12691* (2022).

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. 2022. Maskgit: Masked generative image transformer. *arXiv preprint arXiv:2202.04200* (2022).

Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. 2020. Generative pretraining from pixels. In *International Conference on Machine Learning*. PMLR, 1691–1703.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020a. Stargan v2: Diverse image synthesis for multiple domains. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 8188–8197.

Yunjey Choi, Youngjung Uh, Jaejun Yoo, and Jung-Woo Ha. 2020b. StarGAN v2: Diverse Image Synthesis for Multiple Domains. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Katherine Crowson, Stella Biderman, Daniel Kornis, Dashiell Stander, Eric Hallahan, Louis Castricato, and Edward Raff. 2022. VQGAN-CLIP: Open Domain Image Generation and Editing with Natural Language Guidance. *arXiv preprint arXiv:2204.08583* (2022).

Navneet Dalal and Bill Triggs. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, Vol. 1. Ieee, 886–893.

Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. 2008. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (Csur)* 40, 2 (2008), 1–60.

Prafulla Dhariwal and Alexander Nichol. 2021. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. 2017. Density estimation using Real NVP. In *International Conference on Learning Representations (ICLR)*.

DC Dowson and BV666017 Landau. 1982. The Fréchet distance between multivariate normal distributions. *Journal of multivariate analysis* 12, 3 (1982), 450–455.

Kshitij Dwivedi and Gemma Roig. 2019. Representation similarity analysis for efficient task taxonomy & transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. 2010. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *IEEE transactions on visualization and computer graphics* 17, 11 (2010), 1624–1636.

Ahmed Elgammal. 2019. AI is blurring the definition of artist: Advanced algorithms are using machine learning to create art autonomously. *American Scientist* 107, 1 (2019), 18–22.

Patrick Esser, Robin Rombach, and Bjorn Ommer. 2021. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Fartash Faghri, David J Fleet, Jamie Ryan Kiros, and Sanja Fidler. 2017. Vse++: Improving visual-semantic embeddings with hard negatives. *arXiv preprint arXiv:1707.05612* (2017).

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc'Aurelio Ranzato, and Tomas Mikolov. 2013. Devise: A deep visual-semantic embedding model. *Advances in neural information processing systems* 26 (2013).

Rinon Gal, Or Patashnik, Haggai Maron, Gal Chechik, and Daniel Cohen-Or. 2022. StyleGAN-NADA: CLIP-Guided Domain Adaptation of Image Generators. *ACM Transactions on Graphics (TOG)* (2022).

Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2012. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE transactions on pattern analysis and machine intelligence* 35, 12 (2012), 2916–2929.

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 (NeurIPS)*.

Venkat N Gudivada and Vijay V Raghavan. 1995. Content based image retrieval systems. *Computer* 28, 9 (1995), 18–22.

David Ha and Jürgen Schmidhuber. 2018. World models. *arXiv preprint arXiv:1803.10122* (2018).

Erik Härkönen, Aaron Hertzmann, Jaakko Lehtinen, and Sylvain Paris. 2020. GANSpace: Discovering Interpretable GAN Controls. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Aaron Hertzmann. 2020. Computers do not make art, people do. *Commun. ACM* 63, 5 (2020), 45–48.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017a. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017b. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural computation* 14, 8 (2002), 1771–1800.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Weiming Hu, Nianhua Xie, Li Li, Xianglin Zeng, and Stephen Maybank. 2011. A survey on visual content-based video indexing and retrieval. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 41, 6 (2011), 797–819.

Xun Huang, Arun Mallya, Ting-Chun Wang, and Ming-Yu Liu. 2021. Multimodal Conditional Image Synthesis with Product-of-Experts GANs. *arXiv preprint arXiv:2112.05130* (2021).

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. 2016. What makes ImageNet good for transfer learning? *arXiv preprint arXiv:1608.08614* (2016).

Wonjong Jang, Gwangjin Ju, Yucheol Jung, Jiaolong Yang, Xin Tong, and Seungyong Lee. 2021. StyleCariGAN: Caricature Generation via StyleGAN Feature Map Modulation. *ACM Transactions on Graphics (TOG)* (2021).

Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. 2010. Aggregating local descriptors into a compact image representation. In *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 3304–3311.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*. PMLR, 4904–4916.

Andrej Karpathy, Armand Joulin, and Li F Fei-Fei. 2014. Deep fragment embeddings for bidirectional image sentence mapping. *Advances in neural information processing systems* 27 (2014).

Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. 2018. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations (ICLR)*.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. 2020a. Training Generative Adversarial Networks with Limited Data. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tero Karras, Miika Aittala, Samuli Laine, Erik Härkönen, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2021. Alias-Free Generative Adversarial Networks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Tero Karras, Samuli Laine, and Timo Aila. 2019. A style-based generator architecture for generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. 2020b. Analyzing and improving the image quality of stylegan. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Diederik P Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. 2019. Do better imagenet models transfer better?. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Alex Krizhevsky and Geoffrey E Hinton. 2011. Using very deep autoencoders for content-based image retrieval.. In *ESANN*, Vol. 1. Citeseer, 2.

Nupur Kumari, Richard Zhang, Eli Shechtman, and Jun-Yan Zhu. 2022. Ensembling Off-the-shelf Models for GAN Training. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tuomas Kynkäänniemi, Tero Karras, Miika Aittala, Timo Aila, and Jaakko Lehtinen. 2022. The Role of ImageNet Classes in Fr\'echet Inception Distance. *arXiv preprint arXiv:2203.06026* (2022).

Kathleen M Lewis, Srivatsan Varadharajan, and Ira Kemelmacher-Shlizerman. 2021. TryOnGAN: Body-Aware Try-On via Layered Interpolation. *ACM Transactions on Graphics (TOG)* (2021).

Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. 2019. Photo-sketching: Inferring contour drawings from images. In *Winter Conference on Applications of Computer Vision*.

Yijun Li, Richard Zhang, Jingwan Lu, and Eli Shechtman. 2020. Few-shot image generation with elastic weight consolidation. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yen-Liang Lin, Cheng-Yu Huang, Hao-Jeng Wang, and Winston Hsu. 2013. 3D sub-query expansion for improving sketch-based multi-view image retrieval. In *IEEE International Conference on Computer Vision (ICCV)*.

Huan Ling, Karsten Kreis, Daiqing Li, Seung Wook Kim, Antonio Torralba, and Sanja Fidler. 2021. EditGAN: High-Precision Semantic Image Editing. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Bingchen Liu, Yizhe Zhu, Kunpeng Song, and Ahmed Elgammal. 2021. Towards faster and stabilized gan training for high-fidelity few-shot image synthesis. In *International Conference on Learning Representations (ICLR)*.

Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. 2017. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Ming-Yu Liu, Xun Huang, Jiahui Yu, Ting-Chun Wang, and Arun Mallya. 2020. Generative adversarial networks for image and video synthesis: Algorithms and applications. *arXiv preprint arXiv:2008.02793* (2020).

Yunzhe Liu, Rinon Gal, Amit H Bermano, Baoquan Chen, and Daniel Cohen-Or. 2022. Self-Conditioned Generative Adversarial Networks for Image Editing. In *ACM SIGGRAPH*.

David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *International journal of computer vision* 60, 2 (2004), 91–110.

lucid layers. 2022. Datasets and pretrained Models for StyleGAN3. https://github.com/edstoica/lucid_stylegan3_datasets_models/.

Christopher Manning, Prabhakar Raghavan, and Hinrich Schütze. 2010. Introduction to information retrieval. *Natural Language Engineering* 16, 1 (2010), 100–103.

Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. 2022. SDEdit: Guided Image Synthesis and Editing with Stochastic Differential Equations. In *International Conference on Learning Representations (ICLR)*.

Sangwoo Mo, Minsu Cho, and Jinwoo Shin. 2020. Freeze the Discriminator: a Simple Baseline for Fine-Tuning GANs. In *CVPR Workshop*.

Ron Mokady, Michal Yarom, Omer Tov, Oran Lang, Daniel Cohen-Or, Tali Dekel, Michal Irani, and Inbar Mosseri. 2022. Self-Distilled StyleGAN: Towards Generation from Internet Photos. In *ACM SIGGRAPH*.

Basil Mustafa, Carlos Riquelme, Joan Puigcerver, André Susano Pinto, Daniel Keysers, and Neil Houlsby. 2020. Deep Ensembles for Low-Data Transfer Learning. *arXiv preprint arXiv:2010.06866* (2020).

Atsuhiro Noguchi and Tatsuya Harada. 2019. Image generation from small datasets via batch statistics adaptation. In *IEEE International Conference on Computer Vision (ICCV)*.

Utkarsh Ojha, Yijun Li, Cynthia Lu, Alexei A. Efros, Yong Jae Lee, Eli Shechtman, and Richard Zhang. 2021. Few-shot Image Generation via Cross-domain Correspondence. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Aude Oliva and Antonio Torralba. 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision* 42, 3 (2001), 145–175.

Aaron van den Oord, Nal Kalchbrenner, Oriol Vinyals, Lasse Espeholt, Alex Graves, and Koray Kavukcuoglu. 2016. Conditional image generation with PixelCNN decoders. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. 2021. On Buggy Resizing Libraries and Surprising Subtleties in FID Calculation. *arXiv preprint arXiv:2104.11222* (2021).

Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. 2021. Styleclip: Text-driven manipulation of stylegan imagery. In *IEEE International Conference on Computer Vision (ICCV)*.

William Peebles, Jun-Yan Zhu, Richard Zhang, Antonio Torralba, Alexei Efros, and Eli Shechtman. 2022. GAN-Supervised Dense Visual Alignment. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE conference on computer vision and pattern recognition*. IEEE, 1–8.

Justin Pinkney. 2020a. Awesome Pretrained StyleGAN. https://www.justinpinkney.com/pretrained-stylegan/.

Justin Pinkney. 2020b. Awesome Pretrained StyleGAN2. https://github.com/justinpinkney/awesome-pretrained-stylegan2.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, Cedric Renggli, André Susano Pinto, Sylvain Gelly, Daniel Keysers, and Neil Houlsby. 2021. Scalable transfer learning with expert models. In *International Conference on Learning Representations (ICLR)*.

Filip Radenovic, Giorgos Tolias, and Ondrej Chum. 2018. Deep shape matching. In *European Conference on Computer Vision (ECCV)*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*.

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

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. 2019. Generating diverse high-fidelity images with vq-vae-2. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. 2020. Sketchformer: Transformer-based representation for sketched structure. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Daniel Roich, Ron Mokady, Amit H Bermano, and Daniel Cohen-Or. 2021. Pivotal Tuning for Latent-based Editing of Real Images. *arXiv preprint arXiv:2106.05744* (2021).

Kate Saenko, Brian Kulis, Mario Fritz, and Trevor Darrell. 2010. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*.

Ruslan Salakhutdinov and Geoffrey Hinton. 2009. Semantic hashing. *International Journal of Approximate Reasoning* 50, 7 (2009), 969–978.

Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. 2016. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)* 35, 4 (2016), 1–12.

Axel Sauer, Katja Schwarz, and Andreas Geiger. 2022. Stylegan-xl: Scaling stylegan to large diverse datasets. In *ACM SIGGRAPH*.

Derrick Schultz. 2020. FreaGAN, undertrained GAN trained on Frea Buckler's artwork. https://twitter.com/dvsch/status/1255858574560225284.

Yujun Shen, Ceyuan Yang, Xiaoou Tang, and Bolei Zhou. 2020. InterFaceGAN: Interpreting the Disentangled Face Representation Learned by GANs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* (2020).

Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. 2011. Data-driven visual similarity for cross-domain image matching. In *ACM SIGGRAPH Asia*.

Josef Sivic and Andrew Zisserman. 2003. Video Google: A text retrieval approach to object matching in videos. In *Computer Vision, IEEE International Conference on*, Vol. 3. IEEE Computer Society, 1470–1470.

Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. 2000. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence* 22, 12 (2000), 1349–1380.

Richard Socher, Andrej Karpathy, Quoc V Le, Christopher D Manning, and Andrew Y Ng. 2014. Grounded compositional semantics for finding and describing images with sentences. *Transactions of the Association for Computational Linguistics* 2 (2014), 207–218.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*. PMLR, 2256–2265.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021a. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*.

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021b. Denoising diffusion implicit models. In *International Conference on Learning Representations (ICLR)*.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

A. Tewari, O. Fried, J. Thies, V. Sitzmann, S. Lombardi, K. Sunkavalli, R. Martin-Brualla, T. Simon, J. Saragih, M. Nießner, R. Pandey, S. Fanello, G. Wetzstein, J.-Y. Zhu, C. Theobalt, M. Agrawala, E. Shechtman, D. B Goldman, and M. Zollhöfer. 2020. State of the Art on Neural Rendering. *Computer Graphics Forum (EG STAR 2020)* (2020).

Antonio Torralba, Rob Fergus, and Yair Weiss. 2008. Small codes and large image databases for recognition. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 1–8.

Omer Tov, Yuval Alaluf, Yotam Nitzan, Or Patashnik, and Daniel Cohen-Or. 2021. Designing an Encoder for StyleGAN Image Manipulation. *ACM Transactions on Graphics (TOG)* (2021).

Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. 2021. Sketch Your Own GAN. In *IEEE International Conference on Computer Vision (ICCV)*.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yaxing Wang, Chenshen Wu, Luis Herranz, Joost van de Weijer, Abel Gonzalez-Garcia, and Bogdan Raducanu. 2018. Transferring gans: generating images from limited data. In *European Conference on Computer Vision (ECCV)*.

Yair Weiss, Antonio Torralba, and Rob Fergus. 2008. Spectral hashing. *Advances in neural information processing systems* 21 (2008).

Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. 2020. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks?. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Amir R Zamir, Alexander Sax, William Shen, Leonidas J Guibas, Jitendra Malik, and Silvio Savarese. 2018. Taskonomy: Disentangling task transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuxuan Zhang, Huan Ling, Jun Gao, Kangxue Yin, Jean-Francois Lafleche, Adela Barriuso, Antonio Torralba, and Sanja Fidler. 2021. Datasetgan: Efficient labeled data factory with minimal human effort. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 10145–10155.

Miaoyun Zhao, Yulai Cong, and Lawrence Carin. 2020a. On leveraging pretrained GANs for generation with limited data. In *International Conference on Machine Learning (ICML)*.

Shengyu Zhao, Zhijian Liu, Ji Lin, Jun-Yan Zhu, and Song Han. 2020b. Differentiable Augmentation for Data-Efficient GAN Training. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Liang Zheng, Yi Yang, and Qi Tian. 2017. SIFT meets CNN: A decade survey of instance retrieval. *IEEE transactions on pattern analysis and machine intelligence* 40, 5 (2017), 1224–1244.

Jiapeng Zhu, Ruili Feng, Yujun Shen, Deli Zhao, Zhengjun Zha, Jingren Zhou, and Qifeng Chen. 2021b. Low-Rank Subspaces in GANs. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A Efros. 2016. Generative visual manipulation on the natural image manifold. In *European Conference on Computer Vision (ECCV)*.

Peihao Zhu, Rameen Abdal, John Femiani, and Peter Wonka. 2021a. Barbershop: GAN-based Image Compositing using Segmentation Masks. arXiv:2106.01505 [cs.CV]

Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. 2020. Improved StyleGAN Embedding: Where are the Good Latents? *arXiv preprint arXiv:2012.09036* (2020).