CoGS: Controllable Generation and Search from Sketch and Style

Cusuh Ham[1][*], Gemma Canet Tarrés[2][*], Tu Bui[2],
James Hays[1], Zhe Lin[3], and John Collomosse[2,3]

[1] Georgia Institute of Technology {`cusuh,hays`}`@gatech.edu`
[2] University of Surrey {`g.canettarres,t.v.bui,j.collomosse`}`@surrey.ac.uk`
[3] Adobe Inc. `zlin@adobe.com`

Abstract. We present CoGS, a novel method for the style-conditioned, sketch-driven synthesis of images. CoGS enables exploration of diverse appearance possibilities for a given sketched object, enabling decoupled control over the structure and the appearance of the output. Coarse-grained control over object structure and appearance are enabled via an input sketch and an exemplar "style" conditioning image to a transformer-based sketch and style encoder to generate a discrete codebook representation. We map the codebook representation into a metric space, enabling fine-grained control over selection and interpolation between multiple synthesis options before generating the image via a vector quantized GAN (VQGAN) decoder. Our framework thereby unifies search and synthesis tasks, in that a sketch and style pair may be used to run an initial synthesis which may be refined via combination with similar results in a search corpus to produce an image more closely matching the user's intent. We show that our model, trained on the 125 object classes of our newly created Pseudosketches dataset, is capable of producing a diverse gamut of semantic content and appearance styles.

Keywords: Image Generation, Sketch, Style, Generative Search.

1   Introduction

Generative artwork is a transforming creative practice, driven by advances in deep networks that enable diverse and realistic image synthesis [9,40,54,10,28,59]. Sketches offer an intuitive modality to both control visual synthesis, and to search visual content [4,44,50]. However sketches are inherently ambiguous, offering incomplete descriptions of users' intent [12,11]. While a sketch may communicate a rough approximation of structure or semantic layout, it offers limited fine-grained control over appearance or texture. This presents a barrier to the practical use of sketches as a tool for controlled search and synthesis. Thus, sketches are limited to serendipitous "content discovery" use cases, rather than as a tool to obtain an image with content and appearance matching a users' target intent.

Inspired by the recently proposed DALL-E architecture [40] for diverse, text-driven image synthesis, we introduce CoGS, a novel method for diverse, sketch-driven image synthesis with fine-grained control over structure and appearance (Fig. 1). We make the following technical contributions:
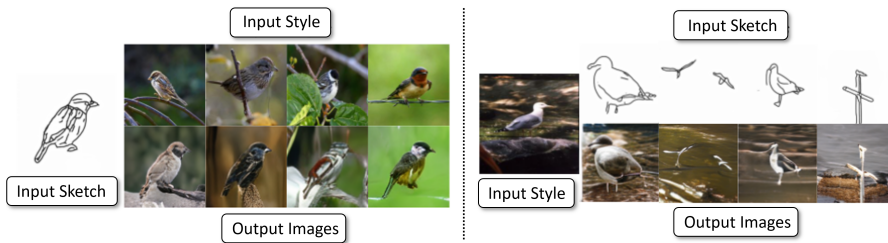
---

[*] Equal contribution.

Fig. 1. Images synthesized using CoGS. On the left we demonstrate the ability to control the style of the output for a given sketch, and on the right we demonstrate the ability to control the structure via multiple sketches for a given style image.

1. Style-conditioned sketch-based synthesis using Vector Quantized GANs (VQGANs) [16]. We synthesize images via a VQGAN decoder, driven by discrete codebook representations generated via a transformer-based sketch and style encoder. We show decoupled coarse-grained control over the structure and appearance of the synthesized image using a provided sketch and exemplar "style" image. This enables users to explore multiple appearance possibilities for a given sketched object, and to do so for a diverse set of object classes.

2. Unified embedding for search and synthesis. We propose a variational auto-encoder (VAE) [36] module that maps codebook representations into a metric space. The embedding thus enables fine-tuning the appearance of the synthesized image either by exploring the local space, or by interpolating between existing similar images within a search corpus.

3. Paired sketch-image dataset. To enable training and evaluation of our model, we create a novel paired dataset of images, hand-drawn sketches, and "pseudosketches" derived from the images via an automated process and graded via crowdsourcing. Specifically, the dataset comprises of 113,370 images and corresponding pseudosketches. A subset of 9,580 images map to the existing Sketchy Database [46], providing 5-10 free-hand sketches for each image.

2   Related Work

Sketch-based image generation was first explored through non-parametric patch-based approaches initially developed for texture synthesis [14,57,2,23], and extended to image synthesis via visual analogy [24] and interactive montage [7]. These methods recall and blend textures sampled from a training set, guided by labeled sketches or semantic maps [3]. With the advent of deep learning, image translation networks, such as CycleGAN [65] and pix2pix [33,56], were exploited to map sketches directly to photos for a single class [58,48,20]. These methods were later adapted for high-resolution portrait synthesis (e.g., via convolutional inversion [21] and semantic priors [60]). SketchyGAN [9] and ContextualGAN [37] proposed multi-class extensions for creating low-resolution outputs of individual objects, building upon the success of conditional generative adversarial networks (cGANs) [38]. Zhu *et al.* proposed a manifold exploration technique to improve controllability of cGAN synthesis [64]. Gao *et al.* showed that these approaches were insufficient for compositing scenes using objects of differing

classes, proposing to fuse scene graph representations with cGANs for direct sketch-to-scene image synthesis [17]. Scene compositions were also produced by mapping sketches to semantic segmentation maps [43] using SPADE [39].

Conditional image generation has been explored for multiple input modalities. Text embeddings [41,42] have been incorporated into both generator and discriminator of cGANs for keyword-driven synthesis, and extended to natural language phrases. A two-stage generator guided via attention from input features was proposed in [54]. Images have been used for conditioning upsampling [18] and inpainting [22], while sketches mostly focus on image colorization [31]. The use of bounding box layouts [53,62,52] and scene graphs [34,1] has also been explored. Recently, transformers have been explored for image translation and completion tasks, and combined with the discrete codebook representations of VQGAN [16]. DALL-E [40] exploited VQGAN to learn a direct mapping from natural language to image for diverse image synthesis. Our work builds upon a similar concept, exchanging text for sketches and further conditioning the codebook generation on an appearance (style). In this sense, our approach is aligned to recent instance-conditioned (IC) synthesis. IC-GAN [6] performed retrieval using descriptors derived from the source image, using an average of spatial semantic maps derived from the top results. Our method optionally leverages retrieval to interpolate between images similar to the synthesized output to fine-tune appearance. Recently, PoE-GAN [28] has also explored multiple guidance cues (sketch, semantic map, text) to synthesize images. Our approach differs both in our transformer architecture and in our two-stage control, offering coarse-grained conditioning on synthesis using a sketch and style image and using a continuous embedding for fine-grained refinement.

3   Methodology

CoGS accepts a labelled sketch (a raster, and an associated semantic class label), and a style image as inputs. Given these three conditioning signals, our goal is to synthesize an image of the specified class that combines the general structure of the sketch with the colors and textures of the style image, providing users with decoupled control over both the content and style of the output image. An optional additional step can be used to further refine the output.

A major challenge for this task is understanding the correspondence between the input sketch and style image. That is, the network must understand how to appropriately map textures from the style image to the corresponding regions of the sketch. We propose to feed an additional input of a class label to resolve correspondence ambiguities, and to clarify the object of interest. CoGS consists of three components (Fig. 2): 1) VQGANs for encoding the sketch and style inputs; 2) a transformer network for sketch- and style-conditioned image synthesis; and 3) a VAE to further refine output through retrieval and interpolation.

3.1   Paired Pseudosketches dataset

While several datasets of human-drawn sketches exist, they are often too small, span only a single or small number of categories [27,51], contain simplistic stock images [51,27,20], or lack paired image correspondences [35,15]. For our work, we require a large-scale dataset of sketch-image pairs across diverse categories in order to learn to synthesize images that capture the structural characteristics
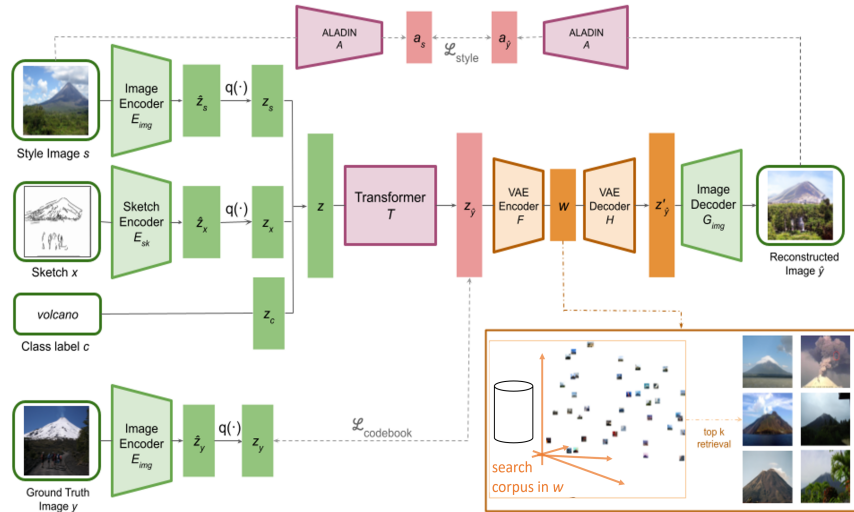
Fig. 2. Illustration of the full CoGS pipeline. We use VQGANs (shown in green) to learn two codebooks, one for sketches and one for images, which are combined with a tokenized representation of a class label. Then, we use a transformer with an auxiliary style loss (shown in red) to learn to composite the inputs into a predicted codebook, which can be subsequently decoded by the image VQGAN decoder to synthesize an image. The input structure and style images offer coarse-grained control over the synthesis. Finally, we use a VAE (shown in orange) to map the codebook to and from a latent space that enables fine-grained refinement of the output image via retrieval or interpolation of results from a search corpus.

of the input sketch. The most relevant existing dataset is the Sketchy Database [47], which contains over 75K sketches of 12.5K images across 125 categories.

The limited number of images in Sketchy DB makes it difficult for a network to learn to synthesize a diverse set of photorealistic outputs. Thus, we propose to create a new dataset of sketch-like edgemaps, or "pseudosketches", extracted from a larger pool of images to create a large set of explicit sketch-image pairs. We take the 12.5K images from Sketchy DB and query additional images from similar ImageNet categories (based on WordNet distances) to run through our automated extraction pipeline shown in Fig. 3.

For each image, we generate a saliency mask using an example-based open-set panoptic segmentation network [29], and blur the non-salient regions with a large Gaussian kern. Then, we run a line drawing extractor [32] to create pseudosketches of the masked images. Blurring is an essential step in making the pseudosketches more similar in nature to hand-drawn sketches. Without blurring, we retain extraneous background details as shown in Fig. 3. Finally, we present the input image and corresponding pseudosketch side-by-side to Amazon Mechanical Turk (AMT) workers, and ask them to rate how representative the pseudosketch is of the image on a scale of 1 (lowest) to 5 (highest), given the semantic class of the image. After filtering out any pseudosketches with a score below 3, we have 113,370 pseudosketch-image pairs across the original 125 categories of Sketchy DB.
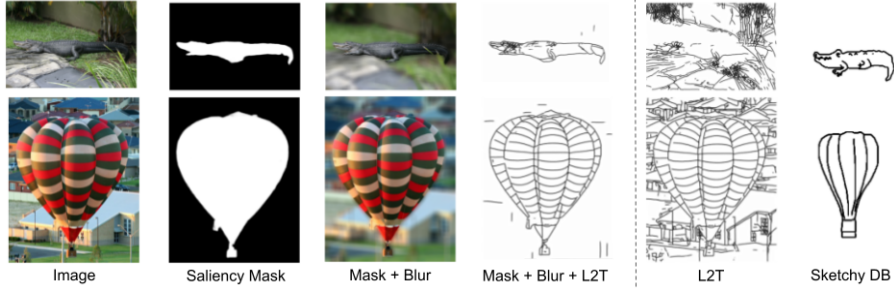
Fig. 3. Stages in the creation of the Pseudosketches dataset. Given an input image, we generate its saliency mask with [29], blur the non-salient regions, and extract the edgemap of the masked image using L2T [32]. Without the saliency mask blurring, we would retain background details not present in hand-drawn sketches.

3.2   Learning effective encodings for input modalities

Given a sketch x, style image s, and class label c as inputs, we need to encode and combine each modality into a single input to the network that synthesizes a composite image. Inspired by recent success with visual sequential modeling [13,40], we want to represent the inputs as a sequence of tokens. We use two vector quantized generative adversarial networks (VQGANs), one trained on pseudosketches and one trained on images, to encode the sketch and style inputs, respectively, into codebook representations. The class label is encoded as a single token representing the index of the class.

Each VQGAN consists of an encoder E, decoder G, and discriminator D. The goal of E and G is to learn the best way to represent an image $x \in \mathbb{R}^{H \times W \times 3}$ as a collection of codebook entries $z_x \in \mathbb{R}^{h \times w \times n_z}$, where n_z is the dimensionality of codes. The discriminator D is trained to ensure that the learned codebook $\mathcal{Z} = \{z_k\}_{k=1}^{K} \subset \mathbb{R}^{n_z}$ is both as rich and compressed as possible, encouraging high quality generative outputs through adversarial training with G.

A quantization step $q(\cdot)$ converts the continuous output of the encoder $\hat{z}_x = E(x) \in \mathbb{R}^{h \times w \times n_z}$ into its discretized form $z_x = q(\hat{z}_x)$ by considering the closest codebook entry z_k to each spatial code of $\hat{z}_x$. Then, G can be trained so that we obtain a reconstruction $\hat{x} = G(z_x)$ as close as possible to the original image x. An additional patch-based discriminator D and perceptual loss are applied to $\hat{x}$ to further ensure that the codebook entries capture details that contribute to improving the overall quality and realism of the image.

3.3   Synthesizing images with transformers

In order to train the CoGS transformer, we must select an image that represents the style connecting each sketch to its corresponding image. We use a pre-trained ALADIN [45] style encoder A to pre-compute the style embedding $a_y = A(y)$ for all images y in the Pseudosketches dataset. Then we compute the pairwise Euclidean distance between all style embeddings $\{a_y^c\}$ belonging to the same class c to find the nearest neighbor s for each y. By limiting the nearest neighbor to come from within the same class, s is guaranteed to be semantically and

stylistically similar to y but not necessarily share the same fine-grained details of the object (e.g., pose, orientation, scale). Thus, the network must learn a more complex relationship between the sketch and style inputs to apply the textures of the style image to the corresponding regions of the sketch.

After determining an appropriate style image for each sketch-image pair, we now have inputs of a sketch x, style image s, and class label c whose combined conditioning signals should produce a target image y. We freeze the weights of the VQGANs described in Section 3.2, and use the encoders E_{sk} and E_{img} to encode the sketch and style images into $\hat{z}_x$ and $\hat{z}_s$, respectively. The quantization step $q(\cdot)$ discretizes the representations into sequences of indices from the codebook $z_x = q(\hat{z}_x)$ and $z_s = q(\hat{z}_s)$ by replacing each code by its index in their respective codebooks, $\mathcal{Z}_{sk}$ and $\mathcal{Z}_{img}$, before concatenating them with the tokenized class encoding z_c into a single input z to a transformer T.

Given a concatenated input z for some input (x, s, c), the transformer T aims to learn $z_y = E_{img}(y)$, which is the codebook representation of the image y corresponding to the input pseudosketch x. The likelihood of the target sequence z_y is modeled autoregressively as

$$p(z_y|x, s, c) = \prod_i p(z_y^i|z_y^{<i}, x, s, c). \tag{1}$$

Thus, the codebook loss is defined as maximizing the log-likelihood:

$$\mathcal{L}_{\text{codebook}} = \mathbb{E}_{y \sim p(y)}[-\log p(z_y|x, s, c)]. \tag{2}$$

We also introduce an auxiliary style loss $\mathcal{L}_{style}$ to further reinforce the style constraint on the generated image. We decode the predicted codebook $z_{\hat{y}} = T(\{z_x, z_s, z_c\})$ using the image VQGAN decoder G_{img} into image $\hat{y} = G_{img}(z_{\hat{y}})$, and compute its ALADIN style embedding $a_{\hat{y}} = A(\hat{y})$ as well as the ALADIN embedding of the input style image $a_s = A(s)$. Then, the style loss is defined as:

$$\mathcal{L}_{\text{style}} = \text{MSE}(a_s, a_{\hat{y}}), \tag{3}$$

where $\text{MSE}(i, j)$ is the mean squared error between i and j. Thus, the total transformer loss is defined as the sum of the two losses weighted by λ_T:

$$\mathcal{L}_{\text{transformer}} = \mathcal{L}_{\text{codebook}} + \lambda_T \mathcal{L}_{\text{style}}. \tag{4}$$

3.4   Refining outputs with VAEs

The decoded image from the transformer's output $\hat{y} = G_{img}(z_{\hat{y}})$ may not always lead to the exact result in mind, whether a consequence of the difficulty of the task or the user wanting to make additional modifications. Therefore, we provide an optional step to further refine the generated image via retrieval or synthesis.

Freezing the VQGANs and transformer from the earlier stages of the pipeline, we train a variational autoencoder (VAE) [36] with encoder F and decoder H to map the encoding $z_{\hat{y}}$ to a more compact representation $w = F(z_{\hat{y}}) \in \mathbb{R}^d$. The VAE is trained on synthesis and search to ensure w, corresponding to the latent space $\mathcal{W} \subset \mathbb{R}^d$, constitutes a unified embedding for both tasks.

The VAE encoder output w is a distribution represented by two vectors: the mean $w_\mu \in \mathbb{R}^d$, and standard deviation $w_\sigma \in \mathbb{R}^d$. The traditional evidence lower

bound (ELBO) objective of VAEs, defined in Eq. (5), encourages a smooth latent space $\mathcal{W}$ that can both reconstruct an input image and synthesize novel outputs:

$$\mathcal{L}_{\text{ELBO}} = \mathcal{L}_{\text{KL}} - \mathcal{L}_{\text{rec}} = \min \mathbb{E}_p[\log p(w|z_{\hat{y}}^q) - \log g(w)] - \mathbb{E}_q \log g(z_{\hat{y}}^q|w). \quad (5)$$

While $\mathcal{L}_{ELBO}$ enables synthesis, we must also enforce metric properties on $\mathcal{W}$ to enable it for retrieval. We leverage self-supervised contrastive learning [8,55,30,26] to map embeddings of structurally similar images close together in latent space and embeddings of dissimilar images far apart:

$$\mathcal{L}_{\text{contrastive}} = -\sum_{i \in I} \log \frac{\exp\left(w_i \cdot w_{j(i)}/\tau\right)}{\sum_{a \in I \setminus i} \exp\left(w_i \cdot w_a/\tau\right)}. \quad (6)$$

For an anchor i in a batch of $2N$ elements, a positive $j(i)$ is an image generated by the CoGS transformer using the same sketch and different style images as i, while the remaining $2(N-1)$ elements are negatives generated using different sketches and completely different style images. Thus, the VAE loss is defined as the sum of the two losses weighted by λ_V:

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{ELBO}} + \lambda_V \mathcal{L}_{\text{contrastive}}. \quad (7)$$

4   Experiments

We describe our experimental setup, evaluation protocols, and comparisons to baseline methods. We evaluate the quality and ability of the transformer component of CoGS to enable both style and structure controllability in Section 4.2-4.6, and investigate the VAE refinement step in Section 4.7.

4.1   Experimental setup

Network architectures and training. We use an ImageNet pre-trained VQ-GAN from [16] for E_{img} and G_{img}, and train another VQGAN on the Pseudosketches dataset for E_{sk}. Both VQGANs are trained with the same settings: the encoder E transforms $x \in \mathbb{R}^{256 \times 256 \times 3}$ to codebooks $z_x \in \mathbb{R}^{16 \times 16 \times 256}$, and the decoder G reconstructs z_x into $\hat{x} \in \mathbb{R}^{256 \times 256 \times 3}$. For the transformer, we use 16 layers with 16 attention heads, and a vocabulary size $|\mathcal{Z}| = 1024$. The dimensionality of the sketch and style codebooks is $n_{\hat{z}_x} = n_{\hat{z}_s} = 256$ and class token $n_{z_c} = 1$, so the concatenated sequences of quantized inputs are of length $n_z = 513$. We randomly partition the Pseudosketches dataset into 102,024 training and 11,346 validation examples, and use a weighting term $\lambda_T = 1$ in Eq. (4).

The VAE encoder F compresses input codebooks $z_y \in \mathbb{R}^{16 \times 16 \times 256}$ into $w \in \mathbb{R}^{1024}$, and the decoder H reconstructs the input back to $z'_y \in \mathbb{R}^{16 \times 16 \times 256}$. With the three components of CoGS being trained sequentially, the VAE is trained using a two-stage approach where training is bootstrapped by only using the contrastive loss $\mathcal{L}_{contrastive}$ for training the encoder F. When the loss converges, both F and H are further trained using the dual loss in Eq. (7) with $\lambda_V = 10^6$.

Evaluation metrics. Two popular metrics to assess the quality of generative networks are the Fréchet Inception Distance (FID) [25] and Learned Perceptual

Image Patch Similarity (LPIPS) [61]. FID computes the distance between the distribution of synthesized validation images and the distribution of the ground truth images corresponding to the input sketches. Thus, the lower the FID, the more similar the generated images are to the distribution of real images.

LPIPS measures the distance between image patches, and is used to quantify a generative network's ability to produce diverse outputs for a set of inputs. We subsample the top 10% validation pseudosketches (as determined by AMT scores during data collection) and their respective style images and class labels, sample 5 output images per input, and average the LPIPS on all unique pairs of outputs.

We further evaluate CoGS using style and structure distance metrics (described in Section 4.4), and AMT to crowd-source quality assessments of CoGS and the baselines. For the latter we source both: 1) user preference for our approach versus baseline approaches, and 2) a subjective evaluation methodology akin to Gao *et al.* [17] that scores how "realistic" each image is, and how "faithful" each synthesized image is to its conditional inputs and ground truth image.

4.2   Dataset partitioning

To approximate the difficulty for the CoGS transformer to synthesize the various categories of the Pseudosketches dataset, we compute the FID on the validation set for each of the 125 classes individually, and group them into 3 sets: the "simple" partition contains classes with the 41 lowest FIDs, "medium" with the 42 middle FIDs, "complex" with the 42 highest FIDs. We then combine the images from all the classes within each partition and recompute the overall FID score of each partition, shown in Table 1. We note that there is a correlation between the number of images in a class and its FID score, i.e., classes with higher number of images tend to produce lower FIDs.

4.3   Comparison to baseline methods

We briefly describe the baseline methods used for evaluations. Other methods such as PoE-GAN [28] and Scribbler [48] align with our work, but lack public code or models for comparison.

Neural Style Transfer (NST) [19] takes as input a content image x_c and style image x_s, and synthesizes an image $\hat{x}$ where the style of x_s is "transferred" onto x_c by optimizing it to match texture (Gram matrix) statistics from x_s using mid-late layer activations of VGG-16 [49]. We use the pseudosketch as the input.

SketchyGAN [9] is a sketch-conditioned GAN trained on Sketchy DB and edgemaps of Flickr images. We substitute Flickr edgemaps for the Pseudosketches dataset to train SketchyGAN as the data and model are not public.

iSketchNFill [20] is a method for inpainting partial sketches and synthesizing images from them. We train the generative component on the Pseudosketches dataset, though the wide class diversity proved challenging for this network.

Instance-Conditioned GAN (IC-GAN) [6] is a conditional GAN that leverages instances of images or text to condition the generated output, guided by pre-computed descriptors from the source images and their top retrieval results. We use the Pseudosketches dataset to train two variants that still compute the descriptors on photorealistic images: 1) add the corresponding pseudosketch's descriptors as an additional conditioning instance, and 2) replace the main photorealistic input with its corresponding pseudosketch.
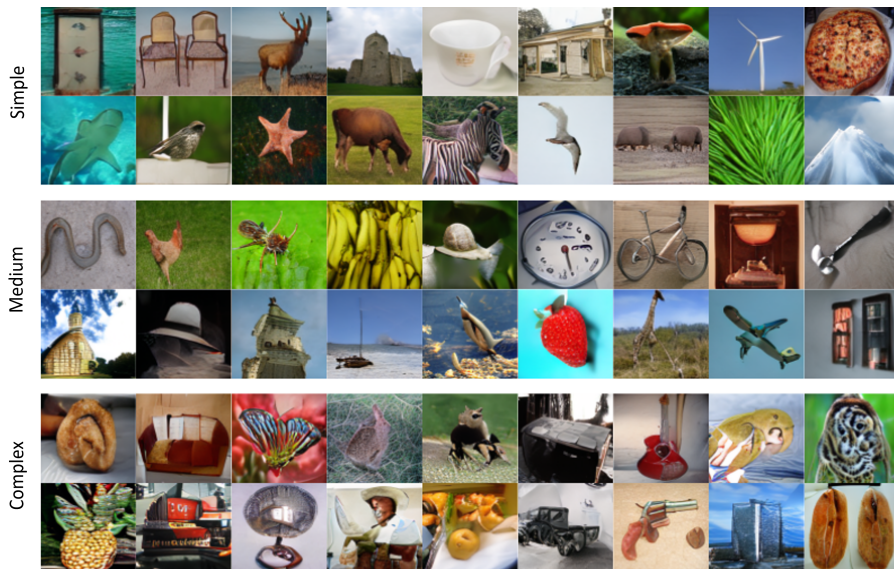
Fig. 4. Representative examples generated by the CoGS transformer stage for the simple, medium, and complex partitions.

CoCosNet v2 [63] is a patch-based method that uses examplar images from different domains, such as edge maps or semantic maps, to translate into high-quality photorealistic images.

Quantitative evaluations using FID and LPIPS are provided in Table 1, and qualitative evaluations based on AMT experiments in Table 2. Our method produces the best FID and comparable LPIPS, indicating good overall quality of images while avoiding mode collapse by producing diverse outputs. The slightly higher LPIPS score from SketchyGAN is likely due to the output being constrained only by the sketch input, whereas the solution space of CoGS is reduced by imposing additional style and class conditions.

We ran three crowd-sourced AMT evaluations: 1) *Realism*: We present output images (e. g. Fig. 5) to participants, and ask them to rate the realism on a scale 1 ("very dissatisfied") to 5 ("very satisfied") using 3 participants per image. We consider only responses with consensus, where the maximum and minimum ratings deviate by at most 1 point; 2) *Fidelity*: We similarly measure the fidelity, or "faithfulness", of the output to the ground truth image. Although scores are generally low (2-3 on the scale), they are similar to scores reported in other papers following this methodology (e.g., [43,17]); and 3) *Preference*: We ask 3 participants per image to indicate their preferred method in terms of overall fidelity to the ground truth image (gt), fidelity of structure to the sketch (sk) and to the style (st). Responses with majority consensus (2 out of 3 agree) are included. CoGS outperforms the baselines in the majority of the AMT studies, and produces the highest ratings overall. NST achieves higher ratings on faithfulness and SketchyGAN on structure only on the complex set.
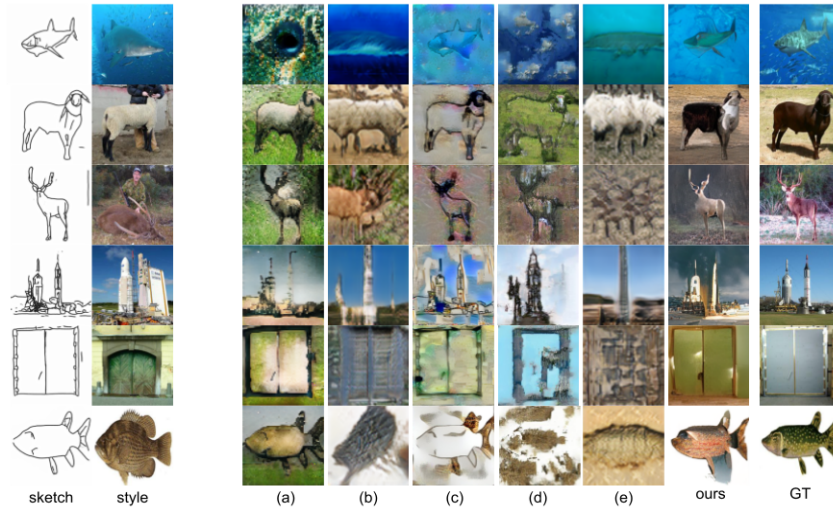
Fig. 5. Representative outputs from: a) SketchyGAN [9], b) IC-GAN [6] (sketch), c) NST [19], d) CoCosNetv2 [63], e) IC-GAN [6] (sketch, style image), and CoGS. Note that neither (a) or (b) use the input style image. We emphasize that the synthesized image may not necessarily aim to look exactly like the "ground truth" image due the stylistic differences between the style and ground truth images (e.g., second row style image contains a white sheep, but the ground truth image is a black sheep).

Method	Simple		Medium		Complex		Overall	
	FID↓	LPIPS↑	FID↓	LPIPS↑	FID↓	LPIPS↑	FID↓	LPIPS↑
SketchyGAN ∘	210.919	**0.534**	256.167	**0.555**	330.649	0.550	213.762	**0.541**
IC-GAN ⬦	103.106	0.183	161.924	0.209	293.037	0.185	109.336	0.190
iSketchNFill ⬦	497.693	1e-8	522.380	9e-9	548.525	1e-8	506.790	1e-8
NST ●	118.839	0.308	167.651	0.302	262.893	0.311	114.707	0.307
CoCosNetv2 ●	153.371	0.399	204.983	0.404	279.311	0.403	160.705	0.401
IC-GAN ⋆	131.750	0.180	176.123	0.216	293.086	0.189	130.235	0.190
CoGS (ours) ⋆	**43.896**	0.500	**95.539**	0.547	**201.230**	**0.616**	**50.630**	0.521

Table 1. Quantitative comparison of FID and LPIPS scores. The symbol next to each method denotes the input(s): $\circ = \{x\}$, $\diamond = \{x, c\}$, $\bullet = \{x, s\}$, and $\star = \{x, s, c\}$, where x is a sketch, s is a style image, and c is the class label.

4.4   Style and structure controllability

While the realism of the generated images is important, the main goal of the CoGS transformer is to provide users decoupled control over both the style and structure of the generated image across a diverse set of categories.

Style controllability. For measuring the stylistic similarity of the generated image to the input style image, we use $d_{style}(s, \hat{y}) = d(a_s, a_{\hat{y}})$, where a_s and $a_{\hat{y}}$ are the style encodings from ALADIN [45] of the style and synthesized images, respectively, and $d(i, j)$ is the Euclidean distance between i and j. In Table 3 we

Method	Simple					Medium					Complex				
	F	R	pref. (gt)	pref. (sk)	pref. (st)	F	R	pref. (gt)	pref. (sk)	pref. (st)	F	R	pref. (gt)	pref. (sk)	pref. (st)
SketchyGAN ∘	2.68	2.67	23.37	36.06	7.04	2.52	2.87	31.72	38.32	10.84	2.64	2.83	36.21	**49.89**	16.19
IC-GAN ◇	2.49	2.52	5.66	0.56	22.26	2.27	2.63	6.04	0.82	21.76	2.36	2.62	3.60	0.67	16.97
iSketchNFill ◇	1.96	1.95	0.00	0.00	0.00	1.87	2.13	0.18	0.74	0.25	1.65	1.99	0.00	0.00	0.26
NST ●	2.69	2.67	10.53	10.73	21.13	2.52	2.85	8.35	7.71	21.76	**2.68**	2.82	10.07	9.89	23.39
IC-GAN ⋆	2.34	2.39	4.32	0.84	15.60	2.14	2.46	4.21	0.67	11.60	2.26	2.51	5.28	0.9	10.80
CoGS (ours) ⋆	**2.94**	**3.04**	**55.93**	**51.80**	**33.96**	**2.67**	**3.05**	**49.44**	**52.40**	**33.78**	2.58	**2.92**	**44.60**	38.65	**32.39**

Table 2. Qualitative evaluations within each partition and across the overall validation set based on AMT experiments. The considered evaluation metrics are: fidelity to ground truth (F), realism of the generated image (R), preference overall given the ground truth image (pref. (gt)), preference based on structure, given the input sketch (pref.(sk)), preference based on style, given the style image (pref.(st)). Values for pref. (gt)/(sk)/(st) correspond to percentages of workers' selection as the best option, while F and R correspond to workers' rankings of outputs from 1 (worst) to 5 (best).

Partition	Style↓	AMT Style↑	Structure↓	AMT Structure↑
Simple	1.085	2.655	2.627	3.117
Medium	1.136	2.374	2.004	2.805
Complex	1.104	2.450	1.519	2.393
Overall	1.100	2.501	2.387	2.776

Table 3. Evaluations for style and structure controllability using distance metrics (ALADIN and Chamfer distances, respectively) and AMT human evaluations where participants are asked to score the fidelity of the output to the input style and sketch on a scale of 1 (low) to 5 (high).

show that the mean style distance of each partition are all within a small epsilon of each other, which agrees with AMT style similarity evaluations, where workers are asked to rate how well the generated image matches the style image on a scale of 1 (worst) to 5 (best). Three participants rate each image, and responses with majority consensus, i.e., min/max scores within 1 point, are included.

We demonstrate style control in Fig. 6. We algorithmically select the 5 style images for each sketch by taking the top $N = 100$ nearest neighbors within the same class in ALADIN style space to the ground truth image corresponding to the input sketch, clustering the N style vectors into $k = 5$ clusters using k-means, and finding the image closest to each of the k centroids. We show that the output images capture the variations in the style image.

Structure controllability. To quantitatively measure the structural fidelity of the generated image to the input sketch, we compute the Chamfer distance $d_{structure}(x, e_{\hat{y}}) = \frac{1}{n}\sum_{i \in x} v_i$, where x is the input sketch, $e_{\hat{y}}$ is the edgemap extracted from the generated image $\hat{y}$ using the Canny edge detector [5], and v_i is the distance transform value of $e_{\hat{y}}$. We only sum over the black pixel coordinates i of x to measure the structural coherence of just the target object. In Table 3 we report the mean structure distance and AMT evaluations, where workers are asked to rate how well the generated image matches the input sketch contours on a scale of 1 (worst) to 5 (best) with consensus.
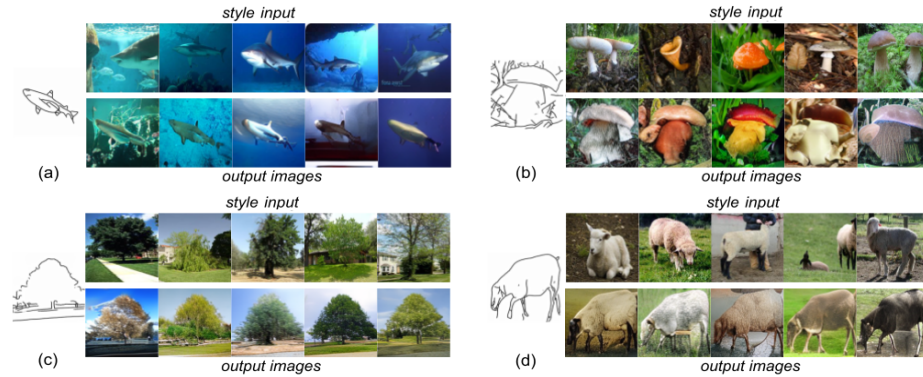
Fig. 6. Style controllability. For each subfigure (a-d), we generate outputs using a single pseudosketch and 5 different style images.
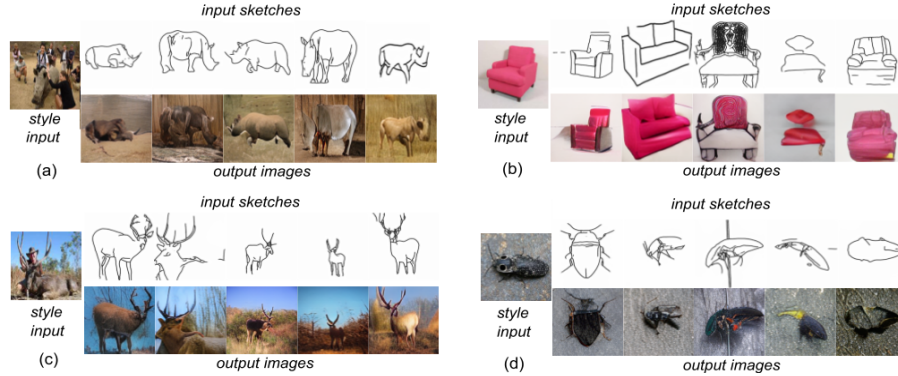

Fig. 7. Structure controllability. For each subfigure (a-d), we generate outputs using 5 pseudosketches for a single style image.

We qualitatively demonstrate the ability to control the structure of the output by sampling the top 10% of pseudosketches for each class (as determined by the AMT score during data collection), and randomly sample one style image from within the same class. We visualize the results in Fig. 7 to show that the synthesized image is guided by the contours of the input sketch.

4.5   Generalization to hand-drawn sketches

Pseudosketches are similarly minimalistic to human-drawn sketches, but their contours have direct pixel correspondences to their ground truth images, whereas human-drawn sketches may come from users of varying skill levels and are more abstract. Although CoGS is only trained on pseudosketches, we show that it is able to generalize to some of the higher quality human-drawn sketches from Sketchy DB (see Fig. 8). Because there is a strong correlation in the number of examples used to compute the FID, we first randomly sample a subset of
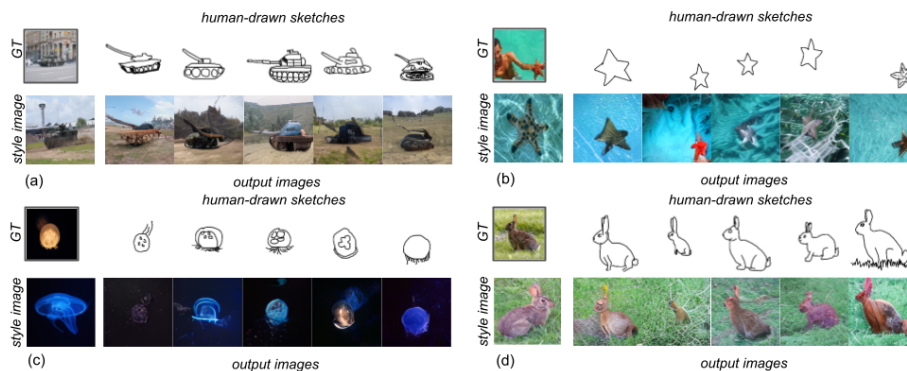
Fig. 8. Generalization. For each subfigure (a-d), we generate outputs using 5 sketches from Sketchy DB [46] corresponding to a ground truth image (framed in grey) with a given style image to demonstrate generalization to hand-drawn sketches.

Sketchy DB similar in size to the Pseudosketches validation set. Our method achieves an FID of 81.820 on Sketchy DB, compared to 50.630 on Pseudosketches, highlighting a gap in the quality of synthesis results from using hand-drawn sketches compared to pseudosketches due to the domain gap and abstraction.

4.6   Transformer ablation study

We investigate the impact of various components of the CoGS transformer using FID, LPIPS, style distance, and structure distance. We show in Table 4 that both the class label c and style loss $\mathcal{L}_{style}$ are important in generating a diverse set of realistic outputs that are consistent with the stylistic and structural conditions.

Inputs	Losses	FID↓	LPIPS↑	Style↓	Structure↓
sketch, style	codebook	58.202	0.519	1.117	3.063
sketch, style	codebook, style	58.327	0.514	1.129	2.823
sketch, style, label	codebook	52.992	0.518	1.120	3.043
sketch, style, label	codebook, style	50.630	0.521	1.100	2.387

Table 4. Ablation study on the inputs and losses for our proposed method.

4.7   Fine-grained control via image retrieval and interpolation

We first evaluate the latent space at offering users similar images, given the transformer output and its class label. We use a VAE trained on the query image class and retrieve the closest validation images from the same class. We show in Fig. 9(a) how, independent of the style and quality of the query image, the retrieved images have the most similar structure. This behavior is validated by
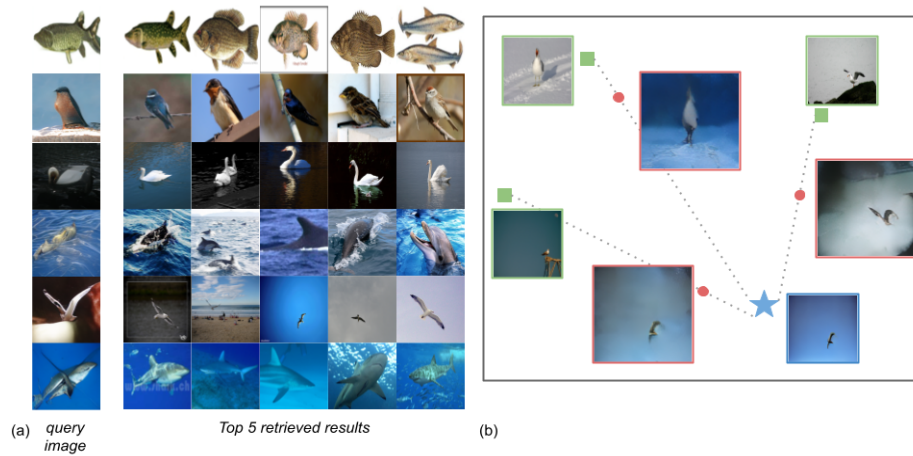
(a) query image   Top 5 retrieved results   (b)

Fig. 9. (a) Top 5 images retrieved from the validation set using generated images as queries. (b) Depiction of the latent space exploration for synthesizing new results. Blue ⋆: query image, green ■: retrieval results, red ●: interpolation results.

AMT evaluations, where 3 participants were asked to rate each retrieved image of a synthesized query image as relevant or not. The majority vote was used to calculate precision@k of 0.533, 0.535, 0.530, 0.520, 0.425 at $k = \{1, 5, 10, 15, 20\}$, respectively, showing how workers agreed on the relevance of up to around the top 15 images, which demonstrates local coherency. Next, we consider the top 3 retrieved images and synthesize new ones by interpolating between each of retrieved images and the query image. We sample 50 images for each interpolation pair, and filter them using an FID threshold of 120 to ensure the quality of the results. We depict the interpolation scheme in Fig. 9(b), showing a few of the many variations the user can create by mixing attributes from two images.

5   Conclusion

We introduce CoGS, a method for image synthesis across a diverse set of categories that provides control over the style and structure of the output image. In order to learn effective controllable synthesis, we collect a large-scale dataset of "pseudosketch"-image correspondences using an automated pipeline. Our approach produces images with higher fidelity to the given style and structure constraints, while also producing diverse and more realistic images within those conditions. We also learn a unified embedding for search and synthesis, which enables further refinement of the generated images via retrieval or interpolation.

Limitations. To broaden the application of our method, we would like to use inputs of hand-drawn sketches from a wide range of skill levels. As shown in Section 4.5, we are able to generalize to examples of hand-drawn sketches with better artistry, but there is still a gap in the FID computed on sketches across skill levels. While this is understandable due to domain gap from the pseudosketches training data, there is room for improvement in resolving abstraction of structure present in hand-drawn sketches.

References

1. Ashual, O., Wolf, L.: Specifying object attributes and relations in interactive scene generation. In: Proc. CVPR (2019)
2. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (ToG) **28**(3), 24 (2009)
3. Barnes, C., Zhang, F.L.: A survey of the state-of-the-art in patch-based synthesis. Computational Visual Media **3**(1), 3–20 (2017)
4. Bui, T., Ribeiro, L., Collomosse, J., Ponti, M.: Sketching out the details: Sketch-based image retrieval using convolutional neural networks with multi-stage regression. Computers —& Graphics **71**, 77 – 87 (2018)
5. Canny, J.: A computational approach to edge detection. IEEE Transactions on pattern analysis and machine intelligence (6), 679–698 (1986)
6. Casanova, A., Careil, M., Verbeek, J., Drozdzal, M., Romero-Soriano, A.: Instance-conditioned gan. arXiv preprint arXiv:2109.05070 (2021)
7. Chen, T., Cheng, M.M., Tan, P., Shamir, A., Hu, S.M.: Sketch2photo: Internet image montage. Proc. ACM SIGGRAPH **28**(5), 124 (2009)
8. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: III, H.D., Singh, A. (eds.) Proceedings of the 37th International Conference on Machine Learning. Proceedings of Machine Learning Research, vol. 119, pp. 1597–1607. PMLR (13–18 Jul 2020), `https://proceedings.mlr.press/v119/chen20j.html`
9. Chen, W., Hays, J.: Sketchygan: Towards diverse and realistic sketch to image synthesis. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
10. Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I., Abbeel, P.: Infogan: Interpretable representation learning by information maximizing generative adversarial nets (Jun 2016)
11. Collomosse, J., Bui, T., Wilber, M., Fang, C., Jin, H.: Sketching with style: Visual search with sketches and aesthetic context. In: Proc. ICCV (2017)
12. Collomosse, J.P., McNeill, G., Watts, L.: Free-hand sketch grouping for video retrieval. In: Proc. ICPR (2008)
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)
14. Efros, A., Freeman, W.: Image quilting for texture synthesis and transfer. In: Proc. SIGGRAPH (2001)
15. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? ACM Trans. Graph. (Proc. SIGGRAPH) **31**(4), 44:1–44:10 (2012)
16. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis (2020)
17. Gao, C., Liu, Q., Xu, Q., Wang, L., Liu, J., Zou, C.: Sketchycoco: Image generation from freehand scene sketches. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
18. Gao, H., Chen, Z., Huang, B., Chen, J., Li, Z.: Image super-resolution based on conditional generative adversarial network. IET Image Processing **14**(13), 3006–3013 (2020)
19. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2414–2423 (2016). https://doi.org/10.1109/CVPR.2016.265
20. Ghosh, A., Zhang, R., Dokania, P.K., Wang, O., Efros, A.A., Torr, P.H.S., Shechtman, E.: Interactive sketch & fill: Multiclass sketch-to-image translation. In: Proceedings of the IEEE international conference on computer vision (2019)

21. Gucluturk, Y., Guclu, U., van Lier, R., van Gerven, M.A.: Convolutional sketch inversion. In: Proc. ECCV Workshop on Vision and Art (VISART) (2016)
22. Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation (2021)
23. Hays, J., Efros, A.A.: Scene completion using millions of photographs. ACM Transactions on Graphics (TOG) **26**(3), 4 (2007)
24. Hertzmann, A., Jacobs, C.E., Oliver, N., Curless, B., Salesin, D.H.: Image analogies. In: Proc. ACM SIGGRAPH. pp. 327–340 (2001)
25. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
26. Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., Bengio, Y.: Learning deep representations by mutual information estimation and maximization. In: International Conference on Learning Representations (2019), `https://openreview.net/forum?id=Bklr3j0cKX`
27. Hospedales, T., Song, Y.Z.: Sketch me that shoe (01 2016)
28. Huang, X., Mallya, A., Wang, T.C., Liu, M.Y.: Multimodal conditional image synthesis with product-of-experts gans (2021)
29. Hwang, J., Oh, S.W., Lee, J., Han, B.: Exemplar-based open-set panoptic segmentation network. CoRR **abs/2105.08336** (2021), `https://arxiv.org/abs/2105.08336`
30. Hénaff, O.J., Razavi, A., Doersch, C., Eslami, S.M.A., Oord, A.v.d.: Data-efficient image recognition with contrastive predictive coding (2019), `http://arxiv.org/abs/1905.09272`, cite arxiv:1905.09272
31. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: Joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Transactions on Graphics (Proc. of SIGGRAPH 2016) **35**(6) (2016)
32. Inoue, N., Ito, D., Xu, N., Yang, J., Price, B., Yamasaki, T.: Learning to trace: Expressive line drawing generation from photographs. Computer Graphics Forum **38**(7), 69–80 (2019). https://doi.org/https://doi.org/10.1111/cgf.13817, `https://onlinelibrary.wiley.com/doi/abs/10.1111/cgf.13817`
33. Isola, P., Zhu, J., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 5967–5976 (2017). https://doi.org/10.1109/CVPR.2017.632
34. Johnson, J., Gupta, A., Fei-Fei, L.: Image synthesis from reconfigurable layout and style. In: Proc. CVPR (2018)
35. Jongejan, J., Rowley, H., Kawashima, T., Kim, J., Fox-Gieg, N.: The quick, draw! a.i. experiment (2016), `https://quickdraw.withgoogle.com/`
36. Kingma, D.P., Welling, M.: Auto-Encoding Variational Bayes. ArXiv e-prints (Dec 2013)
37. Lu, Y., Wu, S., Tai, Y.W., Tang, C.K.: Image generation from sketch constraint using contextual gan. In: The European Conference on Computer Vision (ECCV) (September 2018)
38. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
39. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019)
40. Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., Sutskever, I.: Zero-shot text-to-image generation. arXiv preprint arXiv:2102.12092 (2021)
41. Reed, S., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. In: Advances in Neural Information Processing Systems (NIPS) (2016)

42. Reed, S., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., Lee, H.: Generative adversarial text-to-image synthesis. In: Proc. ICML (2016)
43. Ribeiro, L., Bui, T., Collomosse, J., Ponti, M.: Scene designer: a unified model for scene search and synthesis from sketch. In: Proc. CVPRW on Sketch and Human Expressivity (SHE) (2021)
44. Ribeiro, L.S.F., Bui, T., Collomosse, J., Ponti, M.: Sketchformer: Transformer-based representation for sketched structure. In: Proc. CVPR (2020)
45. Ruta, D., Motiian, S., Faieta, B., Lin, Z.L., Jin, H., Filipkowski, A., Gilbert, A., Collomosse, J.P.: Aladin: All layer adaptive instance normalization for fine-grained style similarity. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 11906–11915 (2021)
46. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM Transactions on Graphics (TOG) **35**(4), 119 (2016)
47. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. ACM Trans. Graph. **35**(4) (jul 2016). https://doi.org/10.1145/2897824.2925954, `https://doi.org/10.1145/2897824.2925954`
48. Sangkloy, P., Lu, J., Fang, C., Yu, F., Hays, J.: Scribbler: Controlling deep image synthesis with sketch and color. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5400–5409 (2017)
49. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
50. Song, J., Song, Y.Z., Xiang, T., Hospedales, T., Ruan, X.: Deep multi-task attribute-driven ranking for fine-grained sketch-based image retrieval. In: British Machine Vision Conference (2016)
51. Song, J., Yu, Q., Song, Y.Z., Xiang, T., Hospedales, T.M.: Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)
52. Sun, W., Wu, T.: Image synthesis from reconfigurable layout and style. In: Proc. CVPR (2019)
53. Sylvain, T., Zhang, P., Bengio, Y., Hjelm, D., Sharma, S.: Object-centric image generation from layouts. arXiv preprint arXiv:2003.07449 (2020)
54. Tang, H., Liu, H., Xu, D., Torr, P., Sebe, N.: Attentiongan: Unpaired image-to-image translation using attention-guided generative adversarial networks. arXiv preprint arXiv:1911.11897 (2019)
55. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. CoRR **abs/1906.05849** (2019), `http://arxiv.org/abs/1906.05849`
56. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2018)
57. Wexler, Y., Shechtman, E., Irani, M.: Space-time video completion. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004. vol. 1, pp. I–I. IEEE (2004)
58. Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F., Hays, J.: Texturegan: Controlling deep image synthesis with texture patches. arXiv preprint arXiv:1706.02823 (2017)
59. Xue, Y., Guo, Y.C., Zhang, H., Xu, T., Zhang, S.H., Huang, X.: Deep image synthesis from intuitive user input: A review and perspectives. Computational Visual Media **8**(1), 3–31 (2022)
60. Yang, Y., Hossain, M.Z., Gedeon, T., Rahman, S.: S2FGAN: Semantically aware interactive sketch-to-face translation. arXiv preprint arXiv:2011.14785 (2020)
61. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)

62. Zhao, B., Meng, L., Yin, W., Sigal, L.: Image generation from layout. In: Proc. CVPR (2019)
63. Zhou, X., Zhang, B., Zhang, T., Zhang, P., Bao, J., Chen, D., Zhang, Z., Wen, F.: Full-resolution correspondence learning for image translation. CoRR **abs/2012.02047** (2020), `https://arxiv.org/abs/2012.02047`
64. Zhu, J.Y., Krahenbuhl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. In: Proc. ECCV (2016)
65. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. arXiv preprint arXiv:1703.10593 (2017)