Unsupervised Sketch to Photo Synthesis

Runtao Liu[1*] Qian Yu[21*] Stella X. Yu[1]

[1]UC Berkeley / ICSI [2]Beihang University

Abstract. Humans can envision a realistic photo given a free-hand sketch that is not only spatially imprecise and geometrically distorted but also without colors and visual details. We study unsupervised sketch to photo synthesis for the first time, learning from *unpaired* sketch and photo data where the target photo for a sketch is unknown during training. Existing works only deal with either style difference or spatial deformation alone, synthesizing photos from edge-aligned line drawings or transforming shapes within the same modality, e.g., color images.
Our insight is to decompose the unsupervised sketch to photo synthesis task into two stages of translation: First shape translation from sketches to grayscale photos and then content enrichment from grayscale to color photos. We also incorporate a self-supervised denoising objective and an attention module to handle abstraction and style variations that are specific to sketches. Our synthesis is sketch-faithful and photo-realistic, enabling sketch-based image retrieval and automatic sketch generation that captures human visual perception beyond the edge map of a photo.

1 Introduction

Sketches, i.e., rapidly executed freehand drawings, make an intuitive and powerful visual expression (Fig.1). There is much research on sketch recognition [7,35],

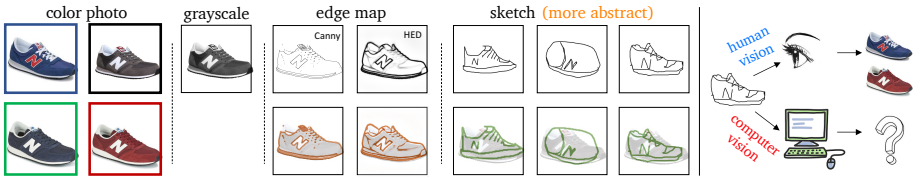

Fig. 1. Comparisons of image types and challenges of sketch to photo synthesis. **Left**: A single object shape could have multiple distinctive colorings yet a common or similar grayscale. Edges extracted by Canny and HED detectors lose colorful details but align well with boundaries in the color photo, whereas sketches are more abstract lines drawn with deformations and style variations. Row 2 shows their lines overlaid on the grayscale photo. **Right**: Human vision can imagine a realistic photo given a free-hand sketch. Our goal is to equip computer vision with the same imagination capability.

*Equal contribution. http://sketch.icsi.berkeley.edu

sketch parsing [26,27], and sketch-based image or video retrieval [36,28,21]. We study how to imagine a realistic photo given a sketch that is spatially imprecise and missing colorful details, by learning from *unpaired* sketches and photos.

Sketch to photo synthesis is challenging for three reasons.

1) Sketches of objects often do not match their shapes in photos, since sketches commonly drawn by amateurs have large spatial and geometrical distortion. Translating a sketch to a photo thus requires shape rectification. However, it is not trivial to rectify shape distortion in a sketch, as line strokes are only suggestive of the actual shapes and locations, and the extent of shape fidelity varies widely between individuals. In Fig.1, the three sketches for the same shoe are very different both overall proportions and local stroke styles.

2) sketches are color-less and lacking details. Drawn in black strokes on white paper, sketches outline mostly object boundaries and characteristic interior markings. To synthesize a photo, shading and colorful textures must be filled in properly. However, it is not trivial to fill in details either. Since a sketch could depict multiple photos, any synthesizer must have the capability to produce not only realistic but also diverse photos for a single sketch.

3) sketches may not have corresponding photos. Free-hand sketches can be created from observation, memory, or pure imagination; they are not so widely available as photos, and those with corresponding photos are even rarer. A few sketch datasets exist in computer vision. TU-Berlin [6] and QuickDraw [11] contain sketches only, with 20,000 and 50 million instances over 250 and 345 categories respectively. Contour Drawing [19] and SceneSketchy [39] have sketch-photo image pairs at the scene level; their sketches are either contour tracings or cartoon-style line drawings, neither representative of real-world free-hand sketches. Sketchy [28] has only 500 sketches paired with 100 photos in each of 125 categories. ShoeV2 and ChairV2 [36] contain 6,648/2,000 and 1,297/400 sketches/photos in a single semantic category of shoes and chairs respectively. To enable data-driven learning of sketch to photo synthesis, we must handle limited sketch data and *unpaired* sketches and photos.

Existing works focus on either shape or color translation alone (Fig.2). **1)** Most image synthesis that deals with shape transfiguration tends to stay in the same visual domain, e.g. changing the picture of a dog to that of a cat [22,15], where visual details are comparable in the color image. **2)** Sketches are a special case of line drawings, and the most studied case of line drawings in computer vision is the edge map extracted automatically from a photo. Such an edge map based drawing to photo synthesis task does not have the spatial deformation problem between sketches and photos, and realistic photos can be synthesized *with* [16,31] or *without* [38] paired training data between drawings and photos. We will show that existing methods fail in sketch to photo synthesis when both shape and color translations are needed simultaneously.

We consider learning sketch to photo synthesis from sketches and photos of the same object category such as *shoes*. There is no pairing information between individual sketches and photos; these two sets can be independently collected.

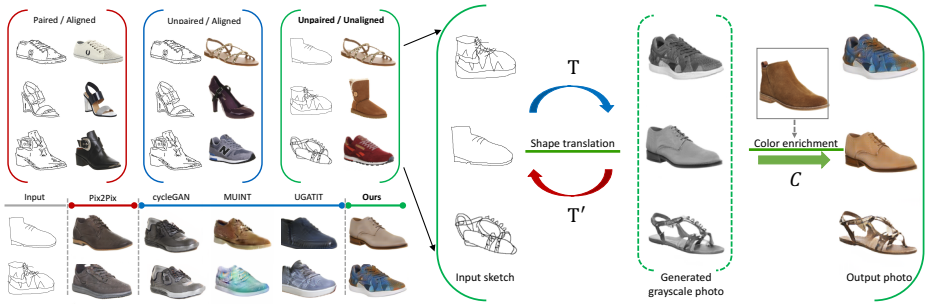

Fig. 2. Comparison of sketch to photo synthesis settings and results. **Left)** Three training scenarios on whether line drawings and photos are provided as paired training instances and whether line drawings are spatially aligned with the photos. Edges extracted from photos are aligned, whereas sketches are not. The bottom panel compares synthesis results from representative approaches in each setting, indicated by the same line/bracket color. Ours are superior to unsupervised edge map to photo methods (cycleGAN [38], MUINT [15], UGATIT [18]) and even supervised methods (Pix2Pix [16]) trained on paired data. **Right)** Our unsupervised sketch-to-photo synthesis model has two separate stages handling spatial deformation and color enrichment respectively: Shape translation learns to synthesize a grayscale photo given a sketch, from unpaired sketch set and photo set, whereas color enrichment learns to fill the grayscale with colorful details given an optional reference photo.

Our insight for unsupervised sketch to photo synthesis is to decompose the task into two separate translations (Fig.2). Our two-stage model performs first shape translation in grayscale and then content fill-in in color. **Stage 1)** Shape translation learns to synthesize a grayscale photo given a sketch, from unpaired sketch set and photo set. Geometrical distortions are eliminated at this step. To handle abstraction and drawing style variations, we apply a self-supervised learning objective to noise sketch compositions, and also introduce an attention module for the model to ignore distractions. **Stage 2)** Content enrichment learns to fill the grayscale with details, including colors, shading, and textures, given an *optional* reference image. It is designed to work with or without reference images. This capability is enabled by a mixed training strategy. Our model can thus produce diverse outputs on demand.

Our model links sketches to photos and can be used directly in sketch-based photo retrieval. Another exciting corollary result from our model is that we can also synthesize a sketch given a photo, even from unseen semantic categories. Strokes in a sketch capture information beyond edge maps defined primarily on intensity contrast and object exterior boundaries. Automatic photo to sketch generation could lead to more advanced computer vision capabilities and serve as a powerful human-user interaction device.

Our work makes the following contributions. **1)** We propose the first two-stage unsupervised model that can generate diverse, sketch-faithful, and photo-realistic images from a single free-hand sketch. **2)** We introduce a self-supervised

learning objective and an attention module to handle abstraction and style variations in sketches. **3)** Our work not only enables sketch-based image retrieval but also delivers an automatic sketcher that captures human visual perception beyond the edge map of a photo. See http://sketch.icsi.berkeley.edu.

2 Related Works

Sketch-based image synthesis. While much progress has been made on sketch recognition [6,35,37] and sketch-based image retrieval [9,13,20,36,28,21], sketch-based image synthesis remains under-explored.

Prior to deep learning (DL), Sketch2Photo [4] and PhotoSketcher [8] compose a new photo from photos retrieved for a sketch. Sketch2Photo [4] first retrieves photos based on the class label, then uses the given sketch to filter them and compose a target photo. PhotoSketcher [8] has a similar pipeline but retrieves photos based on a rather restrictive sketch and hand-crafted features.

The first DL-based free-hand sketch-to-photo synthesis is SketchyGAN [5] , which trains an encoder-decoder model conditioned on the class label for sketch and photo pairs. Contextual GAN [23] treats sketch to photo synthesis as an image completion problem, using the sketch as a weak contextual constraint. Interactive Sketch [10] focuses on multi-class photo synthesis based on incomplete edges or sketches. All of these works rely on paired sketch and photo data and do not address the shape deformation problem.

Sketches are often used in photo editing [1,25,34], e.g., line strokes are drawn on a photo to change the shape of a roof. Unlike our sketch to photo synthesis, these works mainly address a constrained image inpainting problem.

Synthesis from the opposite direction, photo to sketch, has also been studied [29,19]: The former proposes a hybrid model to synthesize a sketch stroke by stroke given a photo, whereas the latter aims to generate boundary-like drawings that capture the outline of the visual scene. Both models require paired data for training. While photo to sketch is not our focus, our model trained only on *shoes* can generate realistic sketches from photos in other semantic categories.

Generative adversarial networks (GAN). GAN has a generator (G) and a discriminator (D): G tries to fake instances that fool D and D tries to detect fakes from reals. GAN is widely used for realistic image generation [24,17] and translation across image domains [16,15].

Pix2Pix [16] is a conditional GAN that maps source images to target images; it requires paired (source,target) data during training. CycleGAN [38] uses a pair of GANs to map an image from the source domain to the target domain and then back to the source domain. Imposing a consistency loss over such a cycle of mappings, it allows both models to be trained together on unpaired source and target images in two different domains. UNIT [22] and MUNIT [15] are variations of CycleGAN, both achieving impressive performance.

None of these methods work well when the source and target images are spatially poorly aligned (Fig.1) and across different appearance domains.

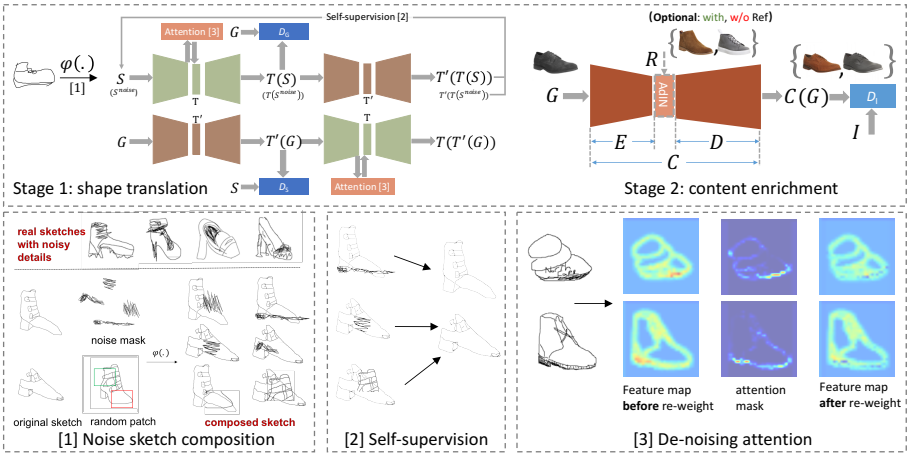

Fig. 3. Our two-stage model architecture (top) and three major technical components (bottom) that tackle abstract and style-varying strokes: noise sketch composition for training data augmentation, a self-supervised de-noising objective, and an attention module to suppress distracting dense strokes.

3 Unsupervised Two-Stage Sketch-to-Photo Synthesis

In our unsupervised learning setting, we are given two sets of data in the same semantic category such as shoes, and no instance pairing is known or available. Formally, all we have are n sketches $\{S_1, \ldots, S_n\}$ and m color photos $\{I_1, \ldots, I_m\}$ along with their grayscale versions $\{G_1, \ldots, G_m\}$.

Compared to photos, sketches are spatially imprecise and colorless. To synthesize a photo from a sketch, we deal with these two aspects at separate stages: We first translate a sketch into a grayscale photo and then translate the grayscale into a color photo filled with missing details on texture and shading (Fig.3).

3.1 Shape Translation: Sketch $S \rightarrow$ Grayscale G

Overview. We first learn to translate sketch S into grayscale photo G. The goal is to rectify shape deformation in sketches. We consider *unpaired* sketch and photo images, not only because paired data are scarce and hard to collect, but also because heavy reliance on paired data could restrict the model from recognizing the inherent misalignment between sketches and photos.

A pair of mappings, $T : S \rightarrow G$ and $T' : G \rightarrow S$, each implemented with an encoder-decoder architecture, are learned with cycle-consistency objectives: $S \approx T'(T(S))$ and $G \approx T(T'(G))$. Similar to [38], we train two domain discriminators D_G and D_S: D_G tries to tease apart G and $T(S)$, while D_S teases apart S and $T'(G)$ (Fig. 3). The predicted grayscale $T(S)$ goes to content enrichment next.

The input sketch may exhibit various levels of abstraction and different drawing styles. In particular, sketches containing dense strokes or noisy details (Fig. 3) cannot be handled well by a basic CycleGAN model.

To deal with these variations, we introduce two strategies for the model to extract style-invariant information only: **1)** We compose additional noise sketches to enrich the dataset and introduce a self-supervised objective; **2)** We introduce an attention module to help detect distracting regions.

Noise sketch composition. In a rapidly drawn sketch, strokes could be deliberately complex, or simply careless and distractive (Fig. 3). We augment limited sketch data with more noise. Let $S^{\text{noise}} = \varphi(S)$, where $\varphi(.)$ represents composition. We detect dense strokes and construct a pool of noise masks. We randomly sample from these masks and artificially generate *complex* sketches by inserting these dense stroke patterns into original sketches. We generate *distractive* sketches by adding a random patch from a different sketch on an existing sketch. The noise strokes and random patches are used to simulate irrelevant details in a sketch. We compose such noise sketches on the fly and feed them into the network with a fixed occurrence ratio.

Self-supervised objective. We introduce a self-supervised objective to work with the synthesized noise sketches. For a composed noise sketch, the reconstruction goal of our model is to reproduce the *original clean* sketch:

$$L_{ss}(T, T') = \left\| S - T'\left(T(S^{\text{noise}})\right) \right\|_1 \tag{1}$$

This objective is different from the cycle-consistency loss used on untouched original sketches. It makes the model ignore irrelevant strokes and put more efforts on style-invariant strokes in the sketch.

Ignore distractions with active attention. To identify distracting strokes, we also introduce an attention module. Since most areas of a sketch are blank, the activation of dense stroke regions is stronger than others. We can thus locate distracting areas and *suppress* the activation there accordingly. That is, the attention module generates an attention map A to be used for re-weighting the feature representation of sketch S (Eq. 2):

$$f_{\text{final}}(S) = (1 - A) \odot f(S) \tag{2}$$

where $f(.)$ refers to the feature map and $\odot$ denotes element-wise multiplication. Our attention is used for area suppression instead of the usual area highlight.

Our total objective for training a shape translation model is:

$$\min_{T,T'} \max_{D_G,D_S} \lambda_1(L_{adv}(T, D_G; S, G) + L_{adv}(T', D_S; G, S)) + \lambda_2 L_{cycle}(T, T'; S, G) + \lambda_3 L_{identity}(T, T'; S, G) + L_{ss}(T, T'; S^{noise}).$$

We follow [38] to add an identity loss $L_{identity}$, which slightly improves the performance. See the details of each loss in the Supplementary.

3.2 Content Enrichment: Grayscale $G \rightarrow$ Color I

Now that we have a predicted grayscale photo G, we learn a mapping C that turns it into color photo I. The goal at this stage is to enrich the generated grayscale photo G with missing appearance details.

Since a color-less sketch could have many colorful realizations, many fill-in's are possible. We thus model the task as a style transfer task and use an *optional* reference color image to guide the selection of a particular style.

We implement C as an encoder (E) and decoder (D) network (Fig. 3). Given a grayscale photo G as the input, the model outputs a color photo I. The input G and the grayscale of the output I, specifically the L-channel in CIE Lab color space of the output should be the same. Therefore we use a self-supervised intensity loss (Eq. 3) to train the model:

$$L_{it}(C) = \|G - \text{grayscale}\,(C\,(G))\|_1 \tag{3}$$

We train discriminator D_I to ensure that I is also as photo-realistic as $I_1, \ldots, I_m$.

To achieve the output diversity, we introduce a conditional module that takes an optional reference image for guidance. We follow AdaIN [14] to inject style information by adjusting the feature map statistics. Specifically, the encoder E takes the input grayscale image G and generates a feature map $\mathbf{x} = E(G)$, then the mean and variance of $\mathbf{x}$ are adjusted by the reference's feature map $\mathbf{x}^{ref} = E(R)$. The new feature map is $\mathbf{x}^{new} = AdaIN(\mathbf{x}, \mathbf{x}^{ref})$ (Eq.4), which is subsequently sent to the decoder D for rendering the final output image I:

$$AdaIN(\mathbf{x}, \mathbf{x}^{\text{ref}}) = \sigma(\mathbf{x}^{\text{ref}})(\frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})}) + \mu(\mathbf{x}^{\text{ref}}) \tag{4}$$

Our model can work with or without reference images, in a *single* network, enabled by a mixed training strategy. When there is no reference image, only intensity loss and adversarial loss are used while $\sigma(\mathbf{x}^{ref})$ and $\mu(\mathbf{x}^{ref})$ are set to 1 and 0 respectively; otherwise, a content loss and style loss are computed additionally. The content loss (Eq. 5) is used to guarantee that the input and output images are consistent perceptually, whereas the style loss (Eq. 6) is to ensure the style of the output is aligned with that of the reference image.

$$L_{cont}(C; G, R) = \|E(D(t)) - t\|_1 \tag{5}$$

$$L_{style}(C; G, R) = \sum_{i=1}^{K} \|\mu\,(\phi_i(D(t))) - \mu\,(\phi_i(R))\|_2 + \sum_{i=1}^{K} \|\sigma\,(\phi_i(D(t))) - \sigma\,(\phi_i(R))\|_2 \tag{6}$$

$$\text{where} \quad t = AdaIN(E(G), E(R)) \tag{7}$$

$\phi_i(.)$ denotes a layer of a pre-trained VGG-19 model. In our implementation, we use $relu1\_1$, $relu2\_1$, $relu3\_1$, $relu4\_1$ layers with equal weights to compute the style loss. Eq. 8 shows the total loss for training the content enrichment model. Network architectures and further details are provided in the Supplementary.

$$\min_{C} \max_{D_I} \lambda_4 L_{adv}(C, D_I; G, I) + \lambda_5 L_{it}(C) + \lambda_6 L_{style}(C; G, R) + \lambda_7 L_{cont}(C; G, R) \tag{8}$$

4 Experiments and Applications

4.1 Experimental Setup and Evaluation Metrics

Datasets. We train our model on two single-category sketch datasets, ShoeV2 and ChairV2 [36], with 6,648/2,000 and 1,297/400 sketches/photos respectively.

Each photo has at least 3 corresponding sketches drawn by different individuals. Note that we do not use pairing information at training. Compared to QuickDraw [11], Sketchy [28], and TU-Berlin [6], sketches in ShoeV2/ChairV2 have more fine-grained details. They demand like-kind details in synthesized photos and are thus more challenging as a testbed for sketch to photo synthesis.

Baselines for image translation. 1) Pix2Pix [16] is our supervised learning baseline which requires paired training data. **2) CycleGAN** [38] is an unsupervised bidirectional image translation model. It is the first to apply cycle-consistency with GANs and allows unpaired training data. **3) MUNIT**[15] is also an unsupervised model that could generate multiple outputs given an input. It assumes that the representation of an image can be decomposed into a content code and a style code. **4) UGATIT** [18] is an attention-based image translation model, with the attention to help the model focus on the domain-discriminative regions and thereby improve the synthesis quality.

Training details. We train our shape translation network for 500(400) epochs on shoes(chairs), and train our content enrichment network for 200 epochs. The initial learning rate is 0.0002, and the input image size is 128×128. We use Adam optimizer with batch size 1. Following the practice by CycleGAN, we train the first 100 epochs at the same learning rate and then linearly decrease the rate to zero until the maximum epoch. We randomly compose *complex* and *distractive* sketches with the possibility of 0.2 and 0.3 respectively. The random patch size is 50×50. When training the content enrichment network, we feed reference images into the network with possibility 0.2.

Evaluation metrics. 1) Fréchet Inception Distance (FID). It evaluates image quality and diversity according to the distance between synthesized and real samples according to the statistics of activations in layer pool3 of a pre-trained Inception-v3. A lower FID value indicates higher fidelity. **2) User study** (Quality). It evaluates subjective impressions in terms of similarity and realism. As in [30], we ask the subject to compare **two** generated photos and select the one better fitting their imagination for a given sketch. We sample 50 pairs for each comparison (more details in Supplementary). **3) Learned perceptual image patch similarity** (LPIPS). It measures the distance between two images. As in [15,38], we use it to evaluate the *diversity* of synthesized photos.

4.2 Sketch-based Photo Synthesis Results

Table 1 shows that: **1)** Our model outperforms all the baselines in terms of FID and user studies. Note that all the baselines adopt one-stage architectures. **2)** All the models perform poorly on ChairV2, probably due to more shape variations but far fewer training data for chairs than for shoes (1:5). **3)** Ours outperforms MUNIT by a large margin, indicating that our task-level decomposition strategy, i.e., two-stage architecture, is more effective than feature-level decomposition for this task. **4)** UGATIT ranks the second on each dataset. It is also an attention-based model, showing the effectiveness of attention in image translation tasks.

Fig. 4. Our model can produce high-fidelity and diverse photos from a sketch. **Top:** Result comparisons. Most baselines cannot handle this task well. While UGATIT can generate realistic photos, our results are more faithful to the input sketch, e.g., the three chair examples. **Bottom:** Results without (Column 2) or with (Column 3) the reference image. Our single content enrichment model can work under both settings, with or without a reference photo (shown in the top right corner).

Table 1. Benchmarks on ShoeV2/ChairV2. '$*$' indicates paired data for training.

	ShoeV2			ChairV2		
Model	FID ↓	Quality ↑	LPIPS ↑	FID ↓	Quality ↑	LPIPS ↑
Pix2Pix*	65.09	27.0	0.071	177.79	13.0	0.096
CycleGAN	79.35	12.0	0.0	124.96	20.0	0.0
MUNIT	92.21	14.5	**0.248**	168.81	6.5	**0.264**
UGATIT	76.89	21.5	0.0	107.24	19.5	0.0
Ours	**48.73**	**50.0**	0.146	**100.51**	**50.0**	0.156

Fig. 5. Left: With different references, our model can produce diverse outputs. **Middle:** Given sketches of similar shoes drawn by different users, our model can capture their commonality as well as subtle distinctions. Each row shows input sketch, synthesized grayscale image, synthesized RGB photo. **Right:** Our model even works for sketches at different completion stages, delivering realistic closely looking shoes.

Comparisons in Fig.4 and Varieties in Fig.5(Left). Our results are more realistic and faithful to the input sketch (e.g., buckle and logo); our synthesis with different reference images produces varieties.

Robustness and Sensitivity in Fig. 5(Middle&Right). We test our ShoeV2 model under two settings: 1) sketches corresponding to the same photo, 2) sketches at different completion stages. Given sketches of similar shoes drawn by different users, our model can capture their commonality as well as subtle distinctions and translate them into photos. Our model also works for sketches at different completion stages (obtained by removing strokes according to their orderings), synthesizing realistic closely-looking shoes for partial sketches.

Generalization across domains in Fig.6(Left). When sketches are randomly sampled from different datasets such as TU-Berlin [6] and Sketchy [28], which have greater shape deformation than ShoeV2, our model trained on ShoeV2 can still produce good results (see more examples in the Supplementary).

Sketches from novel categories in Fig.6(Right). While we focus on a single category training, we nonetheless feed our model sketches from other categories. When the model is trained on shoes, the shape translation network has learned to synthesize a grayscale shoe photo based on a *shoe* sketch. For a non-shoe sketch, our model translates it into a shoe-like photo. Some fine details in the sketch become a common component of a shoe. For example, a car becomes a

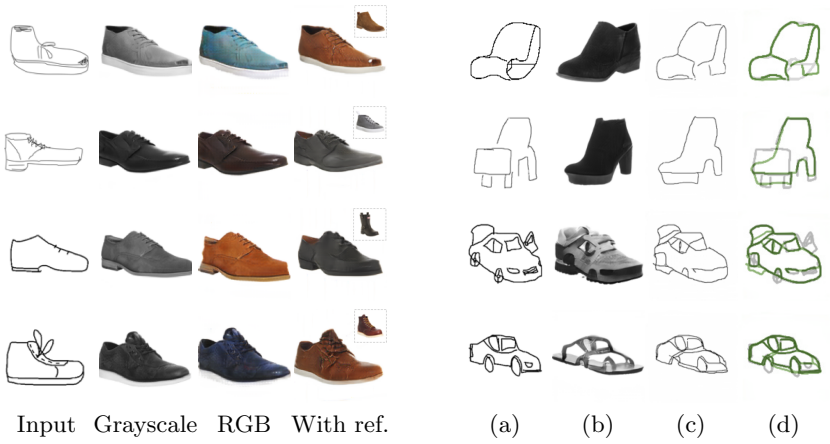

Fig. 6. Left: Generalization across domains. Column 1 are sketches from two unseen datasets, Sketchy and TU-Berlin. Columns 2-4 are results from our model trained on ShoeV2. **Right:** Our shoe model can be used as a shoe detector and generator. It can generate a shoe photo based on a non-shoe sketch. It can further turn the non-shoe sketch into a more shoe-like sketch. (a) Input sketch; (b) synthesized grayscale photo; (c) re-synthesized sketch; (d) Green *(a)* overlaid over gray *(c)*.

Table 2. Comparison of different architecture designs.

FID ↓	CycleGAN(1-stage)	CycleGAN(2-stage)	Edge Map	Grayscale(Ours)
ShoeV2	79.35	51.80	96.58	**48.73**
ChairV2	124.96	109.46	236.38	**100.51**

trainer while the front window becomes part of a shoelace. The superimposition of the input sketch and the re-*shoe*-synthesized sketch reveals which lines are chosen by our model and how it modifies the lines for re-synthesis.

4.3 Ablation Study

Two-stage architecture. Two-stage architecture is the key to the success of our model. This strategy can be easily adapted by other models such as cycleGAN. Table 2 compares the performance of the original cycleGAN and its two-stage version (i.e., cycleGAN is used only for shape translation while the content enrichment network is the same as ours). The two-stage version outperforms the original cycleGAN by 27.55 (on ShoeV2) and 68.33 (on ChairV2), indicating the significant benefits brought by this architectural design.

Edge map vs. grayscale as the intermediate goal. We choose *grayscale* as our intermediate goal of translation. As shown in Fig. 1, *edge maps* could be an alternative since it does not have shape deformation either. We can first translate sketch to an edge map, and then fill the edge map with colorful details.

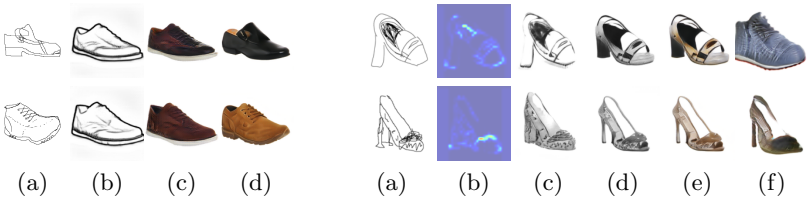

Fig. 7. Left: Synthesized results when the edge map is used as the intermediate goal instead of the grayscale photo. (a) Input sketch; (b) Synthesized edge map, (c) Synthesized RGB photo using the edge map; (d) Synthesized RGB photo using grayscale (Ours). **Right:** Our model can successfully deal with noise sketches, which are not well handled by another attention-based model, UGATIT. For an input sketch (a), our model produce an attention mask (b); (c) and (d) are grayscale images produced by vanilla and our model. (e) and (f) compare ours with the result of UGATIT.

Fig. 8. Comparisons of paired and unpaired training for shape translation. There are four examples. For each example, the 1st one is the input sketch, the 2nd and the 3rd are grayscale images synthesized by Pix2Pix and our model respectively. Note that for each example, although the input sketches are different visually, Pix2Pix produces a similar-looking grayscale image. Our results are more faithful to the sketch.

Table 2 and Fig. 7 show that using the edge map is worse than using the grayscale. Our explanations are: **1)** Grayscale images contain more visual details thus can provide more learning signals for training shape translation network; **2)** Content enrichment is easier for grayscale as they are closer to color photos than edge maps. The grayscale is also easier to obtain in practice.

Deal with abstraction and style variations. We have discussed the problem encountered during shape translation in Section 3.1, and further introduced 1) a self-supervised objective along with noise sketch composition strategies and 2) an attention module to handle the problem. Table 3 compares FID achieved at the first stage by different variants. Our full model can tackle the problem better than the vanilla model, and each component contributes to the improved performance. Figure 7 shows two examples and compares the results of UGATIT.

Paired vs. unpaired training. We train a Pix2Pix model for shape translation to see if paired information helps. As shown in Table 3(*Pix2Pix*) and Fig. 8, It turns out the performance of Pix2Pix is much worse than ours (FID: 75.84 vs. 46.46 on ShoeV2 and 164.01 vs. 90.87 on ChairV2). It is most likely caused by the shape misalignment between sketches and grayscale images.

Exclude the effect of paired information. Although pairing information is not used during training, they do exist in ShoeV2. To eliminate any potential pairing facilitation, we train another model on a composed dataset, created by merging all the sketches of ShoeV2 and 9,995 photos of UT Zappos50K [33].

Table 3. Contribution of each proposed component. The FID scores are obtained based on the results of *shape translation stage*.

FID ↓	Pix2Pix	Vanilla	w/o Self-Supervision	w/o Attention	Ours
ShoeV2	75.84	48.30	46.88	47.0	**46.46**
ChairV2	164.01	104.0	93.33	92.03	**90.87**

Table 4. Exclude the effect of paired data. Although the paired information is not used during training, they indeed exist in ShoeV2. We compose a new dataset where pairing does not exist to train the model again. Results obtained on the same test set.

Dataset	Paired Exist?	Use Pair Info.	FID ↓
ShoeV2	Yes	No	**48.7**
UT Zappos50K	No	No	**48.6**

These photos are collected from a different source than ShoeV2. We train this model in the same setting. In Table 4, we can see this model achieves similar performance with the one trained on ShoeV2, indicating the effectiveness of our approach for learning the task from entirely **unpaired** data.

4.4 Photo-to-Sketch Synthesis Results

Synthesize a sketch given a photo. As the shape translation network is bidirectional (i.e., T and T'), our model can also translate a photo into a sketch. This task is not trivial, as users can easily detect a fake sketch based on its stroke continuity and consistency. Fig. 9(Top) shows that our generated sketches mimic manual line-drawings and emphasize contours that are perceptually significant.
Sketch-like edge extraction. Sketch-to-photo and photo-to-sketch synthesis are opposite processes. We suspect that our model can create sketches from photos in broader categories as it may require less class priors.

We test our shoe model directly on photos in ShapeNet [3]. Figure 9(Bottom) lists our results along with those from HED [32] and Canny edge detector [2]. We also compare with Photo-Sketching [19], a method specifically designed for generating boundary-like drawing from photos. 1) Unlike HED and Canny producing an edge map faithful to the photo, ours presents a hand-drawn style. 2) Our model can dub as an edge+ extractor on unseen classes. This is an exciting corollary product: A promising automatic sketch generator that captures human visual perception beyond the edge map of a photo (more results in Supp.).

4.5 Application: Unsupervised Sketch-based Image Retrieval

Sketch-based image retrieval is an important application of sketch. One of its main challenges is the large domain gap. Existing methods either map sketches and photos into a common space or use edge maps as the intermediate representation. However, our model enables direct mapping between these two domains.

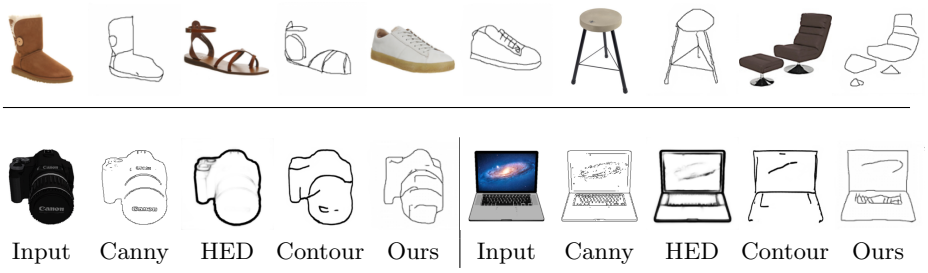

Fig. 9. Our results on photo-based sketch synthesis. **Top**: each sketch-photo pair: left: input photo, right: synthesized sketch. Results obtained on ShoeV2 and ChairV2. **Bottom**: Results obtained on ShapeNet [3]. The column 1 is the input photo, Column 2-5 are lines generated by Canny, HED, Photo-Sketching[19] (*Contour* for short), and our model. Our model can generate line strokes with a hand-drawn effect, while HED and Canny detectors produce edge maps faithful to the original photos. Ours emphasize perceptually significant contours, not intensity-contrast significant as in edge maps.

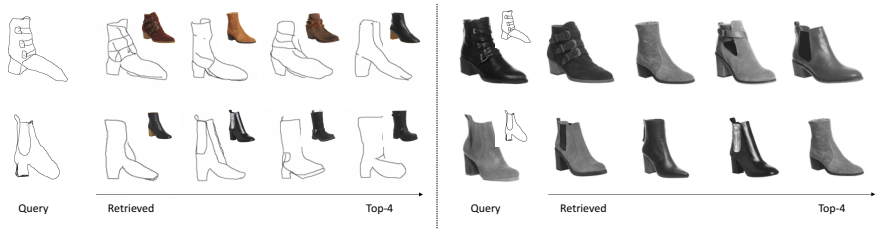

Fig. 10. Sample retrieval results. Our synthesis model can map photo to sketch domain and vice versa. Cross-domain retrieval task can thus be converted to intra-domain retrieval. **Left**: All candidate photos are mapped to sketches, thus both query and candidates are in the sketch domain. **Right**: The query sketch is translated to a photo, so the matching is in the photo domain. Top right shows the original photo or sketch.

We thus conduct experiments in two possible mapping directions: **1)** Translate gallery photos to sketches, and then find the nearest sketches to the query sketch (Fig. 10(Left)); **2)** Translate a sketch to a photo and then find its nearest neighbors in the photo gallery (Fig. 10(Right)). Two ResNet18 [12] models, one is pretrained on the ImageNet while the other is on the TU-Berlin dataset, are used as feature extractors for photos and sketches respectively (see Supplementary for further details). Figure 10 shows our retrieval results. Even *without* any supervision, the results are already acceptable. In the second experiment, we achieve an accuracy of 37.2%(65.2%) at top5 (top20) respectively. These results are higher than the results from *sketch to edge map*, which are 34.5%(57.7%).

Summary. We propose the first unsupervised two-stage sketch-to-photo synthesis model that can produce photos of high fidelity, realism, and diversity. It enables sketch-based image retrieval and automatic sketch generation that captures human visual perception beyond the edge map of a photo.

References

1. Bau, D., Strobelt, H., Peebles, W., Wulff, J., Zhou, B., Zhu, J.Y., Torralba, A.: Semantic photo manipulation with a generative image prior. ACM Transactions on Graphics (TOG) **38**(4), 59 (2019)
2. Canny, J.: A computational approach to edge detection. TPAMI **6**, 679–698 (1986)
3. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
4. Chen, T., Cheng, M.M., Tan, P., Shamir, A., Hu, S.M.: Sketch2photo: internet image montage. In: ACM Transactions on Graphics (TOG) (2009)
5. Chen, W., Hays, J.: Sketchygan: Towards diverse and realistic sketch to image synthesis. In: CVPR (2018)
6. Eitz, M., Hays, J., Alexa, M.: How do humans sketch objects? In: ACM Transactions on Graphics (TOG) (2012)
7. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: An evaluation of descriptors for large-scale image retrieval from sketched feature lines. Computers & Graphics **34**(5), 482–498 (2010)
8. Eitz, M., Richter, R., Hildebrand, K., Boubekeur, T., Alexa, M.: Photosketcher: interactive sketch-based image synthesis. IEEE Computer Graphics and Applications (2011)
9. Eitz, M., Hildebrand, K., Boubekeur, T., Alexa, M.: Sketch-based image retrieval: Benchmark and bag-of-features descriptors. TVCG **17**(11), 1624–1636 (2011)
10. Ghosh, A., Zhang, R., Dokania, P.K., Wang, O., Efros, A.A., Torr, P.H., Shechtman, E.: Interactive sketch & fill: Multiclass sketch-to-image translation. In: CVPR (2019)
11. Ha, D., Eck, D.: A neural representation of sketch drawings. arXiv preprint arXiv:1704.03477 (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hu, R., Barnard, M., Collomosse, J.: Gradient field descriptor for sketch based retrieval and localization. In: ICIP (2010)
14. Huang, X., Belongie, S.: Arbitrary style transfer in real-time with adaptive instance normalization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1501–1510 (2017)
15. Huang, X., Liu, M.Y., Belongie, S., Kautz, J.: Multimodal unsupervised image-to-image translation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 172–189 (2018)
16. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
17. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. arXiv preprint arXiv:1710.10196 (2017)
18. Kim, J., Kim, M., Kang, H., Lee, K.: U-GAT-IT: unsupervised generative attentional networks with adaptive layer-instance normalization for image-to-image translation. CoRR **abs/1907.10830** (2019)
19. Li, M., Lin, Z., Mech, R., Yumer, E., Ramanan, D.: Photo-sketching: Inferring contour drawings from images. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV) (2019)

20. Li, Y., Hospedales, T., Song, Y.Z., Gong, S.: Fine-grained sketch-based image retrieval by matching deformable part models. In: BMVC (2014)
21. Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: Fast free-hand sketch-based image retrieval. arXiv preprint arXiv:1703.05605 (2017)
22. Liu, M.Y., Breuel, T., Kautz, J.: Unsupervised image-to-image translation networks. In: Advances in neural information processing systems. pp. 700–708 (2017)
23. Lu, Y., Wu, S., Tai, Y.W., Tang, C.K.: Image generation from sketch constraint using contextual gan. In: ECCV (2018)
24. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014)
25. Portenier, T., Hu, Q., Szabo, A., Bigdeli, S.A., Favaro, P., Zwicker, M.: Faceshop: Deep sketch-based face image editing. ACM Transactions on Graphics (TOG) **37**(4), 99 (2018)
26. Qi, Y., Guo, J., Li, Y., Zhang, H., Xiang, T., Song, Y.: Sketching by perceptual grouping. In: ICIP. pp. 270–274 (2013)
27. Qi, Y., Song, Y.Z., Xiang, T., Zhang, H., Hospedales, T., Li, Y., Guo, J.: Making better use of edges via perceptual grouping. In: CVPR (2015)
28. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: Learning to retrieve badly drawn bunnies. In: SIGGRAPH (2016)
29. Song, J., Pang, K., Song, Y.Z., Xiang, T., Hospedales, T.M.: Learning to sketch with shortcut cycle consistency. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 801–810 (2018)
30. 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. pp. 8798–8807 (2018)
31. Xian, W., Sangkloy, P., Agrawal, V., Raj, A., Lu, J., Fang, C., Yu, F., Hays, J.: Texturegan: Controlling deep image synthesis with texture patches. In: CVPR (2018)
32. Xie, S., Tu, Z.: Holistically-nested edge detection. In: ICCV (2015)
33. Yu, A., Grauman, K.: Fine-grained visual comparisons with local learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 192–199 (2014)
34. Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Free-form image inpainting with gated convolution. arXiv preprint arXiv:1806.03589 (2018)
35. Yu, Q., Yang, Y., Song, Y., Xiang, T., Hospedales, T.: Sketch-a-net that beats humans. In: BMVC (2015)
36. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: CVPR (2016)
37. Yu, Q., Yang, Y., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M.: Sketch-a-net: A deep neural network that beats humans. IJCV **122**(3), 411–425 (2017)
38. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2223–2232 (2017)
39. Zou, C., Yu, Q., Du, R., Mo, H., Song, Y.Z., Xiang, T., Gao, C., Chen, B., Zhang, H.: Sketchyscene: Richly-annotated scene sketches. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 421–436 (2018)