# Learning Geometry-aware Representations by Sketching

Hyundo Lee, Inwoo Hwang, Hyunsung Go, Won-Seok Choi, Kibeom Kim, Byoung-Tak Zhang
AI Institute, Seoul National University
{hdlee, iwhwang, hsgo, wchoi, kbkim, btzhang}@bi.snu.ac.kr

## Abstract

*Understanding geometric concepts, such as distance and shape, is essential for understanding the real world and also for many vision tasks. To incorporate such information into a visual representation of a scene, we propose learning to represent the scene by sketching, inspired by human behavior. Our method, coined Learning by Sketching (LBS), learns to convert an image into a set of colored strokes that explicitly incorporate the geometric information of the scene in a single inference step without requiring a sketch dataset. A sketch is then generated from the strokes where CLIP-based perceptual loss maintains a semantic similarity between the sketch and the image. We show theoretically that sketching is equivariant with respect to arbitrary affine transformations and thus provably preserves geometric information. Experimental results show that LBS substantially improves the performance of object attribute classification on the unlabeled CLEVR dataset, domain transfer between CLEVR and STL-10 datasets, and for diverse downstream tasks, confirming that LBS provides rich geometric information.*

## 1. Introduction

Since geometric principles form the bedrock of our physical world, many real-world scenarios involve geometric concepts such as position, shape, distance, and orientation. For example, grabbing an object requires estimating its shape and relative distance. Understanding geometric concepts is also essential for numerous vision tasks such as image segmentation, visual reasoning, and pose estimation [27]. Thus, it is crucial to learn a visual representation of the image that can preserve such information [76], which we call *geometry-aware representation*.

However, there is still a challenge in learning geometry-aware representations in a compact way that can be useful for various downstream tasks. Previous approaches have focused on capturing geometric features of an image in a 2D grid structure, using methods such as handcrafted feature extraction [2, 4, 14, 31], segmentation maps [21, 57], or

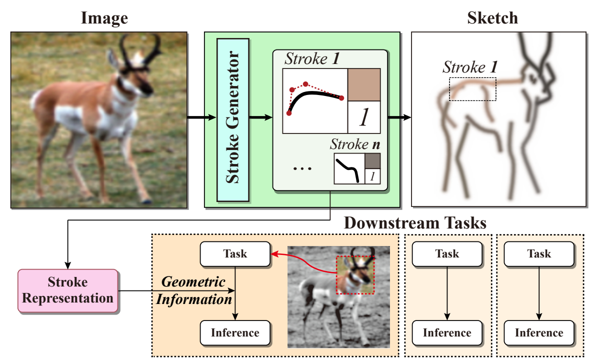


Figure 1. Overview of LBS, which aims to generate sketches that accurately reflect the geometric information of an image. A sketch consists of a set of strokes represented by a parameterized vector that specifies curve, color, and thickness. We leverage it as a geometry-aware representation for various downstream tasks.

convolution features [36, 55]. Although these methods are widely applicable to various domains, they often lack compactness based on a high-level understanding of the scene and tend to prioritize nuisance features such as the background. Another line of work proposes architectures that guarantee to preserve geometric structure [12, 13, 23, 58, 73] or disentangle prominent factors in the data [8, 28, 39, 51]. Although these methods can represent geometric concepts in a compact latent space, they are typically designed to learn features that are specific to a particular domain and often face challenges in generalizing to other domains [64].

In this study, we present a novel approach to learning geometry-aware representations via *sketching*. Sketching, which is the process of converting the salient features of an image into a set of color-coded strokes, as illustrated in Figure 1, is the primary means by which humans represent images while preserving their geometry. Our key idea is that sketches can be a compact, high-level representation of an image that accurately reflects geometric information. Sketching requires a high-level understanding of the scene, as it aims to capture the most salient features of the image and abstract them into a limited number of strokes. In addition, a sketch can be represented as a set of parameters by replacing each stroke with parametric curves. Sketch-

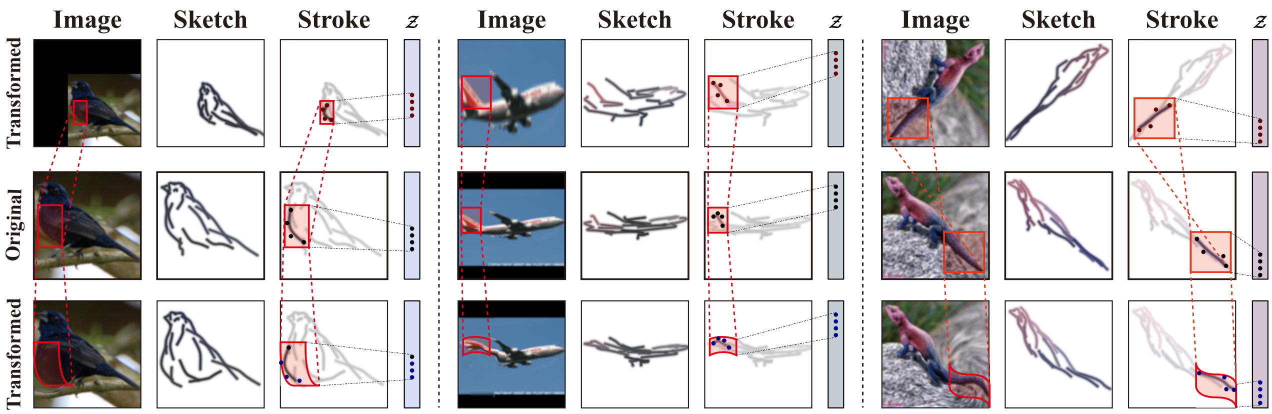


Figure 2. Examples of how sketches capture essential geometric concepts such as shape, size, orientation, curvature, and local distortion. The control points of each stroke compactly represent the local geometric information of the corresponding region in the sketch. The strokes as a whole maintain the geometric structure of the entire image under various transformations in the image domain.

ing has also been linked to how people learn geometric concepts [66]. Based on these properties, we directly use strokes as a geometry-aware representation and utilize them for downstream tasks. Under the theoretical framework of geometric deep learning [3], we conduct theoretical analysis to validate the effectiveness of strokes as a representation and prove their ability to preserve affine transformations.

To validate our hypothesis, we introduce Learning by Sketching (LBS), a method that generates abstract sketches coherent with the geometry of an input image. Our model is distinct from existing sketch generation methods as it does not require a sketch dataset for training, which often has limited abstraction or fails to accurately reflect geometric information. Instead, LBS learns to convert an image into a set of colored Bézier curves that explicitly represent the geometric concepts of the input image. To teach the style of sketching, we use perceptual loss based on the CLIP model [59, 68], which measures both semantic and geometric similarities between images and generated sketches. We propose a progressive optimization process that predicts how strokes will be optimized from their initial positions through CLIP-based perceptual loss to generate abstract sketches in a single inference step. As a result, LBS generates a representation that reflects visual understanding in a single inference step, without requiring a sketch dataset. This produces highly explainable representations through sketches, as illustrated in Figure 2.

We conduct experimental analyses to evaluate the effectiveness of our approach, learning by sketching, by addressing multiple research questions in various downstream tasks, including: **(i)** describing the relationships between geometric primitives, **(ii)** demonstrating simple spatial reasoning ability by conveying global and local geometric information, **(iii)** containing general geometric information shared across different domains, and **(iv)** improving performance on FG-SBIR, a traditional sketch task.

## 2. Related work

### 2.1. Geometry-aware representation learning

Previous studies that aim to capture geometric information within the data can be categorized into two approaches: Firstly, explicitly capturing the geometric information with features on the image space, and secondly, inducing a compact feature vector that preserves the geometric structure of the data.

**Features on image space.** The most straightforward approach to representing geometric structures in images is to establish features directly on the image space, *i.e.*, to define features on the 2D grid structure. Approaches that utilize this method include those that employ traditional edge detection algorithms [4, 35, 74, 75], low-level convolutional features [36, 68], and segmentation maps [49, 57, 61]. However, these features are unsuitable for conversion into compact vector representations for downstream tasks. Alternative approaches include utilizing hand-crafted feature detection algorithms [2, 14, 31, 62], clustering specific areas of an image into superpixels [21, 32], and representing the relationships between specific areas as a graph [15, 65]. Our approach is primarily notable for its ability to capture the most salient image features through an overall understanding of the scene, resulting in a concise representation.

**Geometric deep learning.** Research has also been conducted to develop informative and compact representations that preserve geometric structures in a latent vector space. One representative method is geometric deep learning [3], which involves preserving the geometric properties of an image by learning an invariant (*i.e.*, maintaining the output identically) or an equivariant representation (*i.e.*, there exists an operator acting on the feature space corresponding to the transformations) with respect to geometric transformations. Steerable networks [12, 13, 23, 73] extend the translation equivariance of CNNs to a larger group of symmetries and introduce an efficient weight-sharing architecture. However, designing their architecture requires complex analytic solutions, and they are constrained to particular transformations [37]. Another approach is to disentangle factors of variation into distinct latent subspaces [8, 28, 39, 40, 51, 58, 70]. However, these methods face difficulties in handling changing factors of variation [64] and therefore struggle to generalize to arbitrary geometric transformations. Sketching, in contrast to these methods, can naturally ensure equivariance with respect to geometric transformations by being defined on the image space. In Sec. 3 we show that strokes can be equivariant with respect to arbitrary affine transformations.

### 2.2. Sketch generation methods

Human-drawn sketches are widely used as valuable descriptions in various applications, including image retrieval [16, 47, 78], modeling 3D shapes and surfaces [25, 42, 43, 45], and image editing [33, 56, 69, 71]. Unfortunately, previous works that aim to learn sketching or drawing mostly endeavor to generate artistic works, rather than focusing on its applicability [19, 24, 30, 48, 52, 83].

Sketch generation models often rely on explicit sketch datasets to follow the style of human drawings [5, 10, 46, 60, 67, 82]. However, many existing datasets are not designed to preserve geometric information [7, 20, 63] or contain sketches without sophisticated image abstraction, resembling simple edge detection [1, 72]. This presents challenges for sketch models to accurately abstract an image while preserving its geometries. Without sketch datasets, stroke-based rendering methods usually minimize the reconstruction loss which is related to the pixel-wise distance, which promotes learning in the style of *painting* [19, 30, 48, 52, 53, 81, 83]. There have also been studies that attempt to capture the geometric and semantic features of a scene through imitating the style of line drawing [6, 44, 77]. However, these methods require numerous strokes to cover the whole canvas, primarily focusing on the superfluous background instead of salient features.

**Abstract sketches for representation.** Only a few works aim to generate sketches for the sake of representing an image with a limited number of strokes. Learning to Draw (LtD) [54] creates a representative sketch of an image and uses it as a communication channel between agents. While it primarily focuses on communicating with a specific receiver, we aim to leverage strokes as a general representation for various downstream tasks. CLIPasso [68] utilizes the capabilities of CLIP [59], a neural network trained on images paired with text, to synthesize abstract sketches by maintaining both semantic and geometric attributes. Generating representations with this method, however, is impractical due to its reliance on an optimization-based approach that is extremely time consuming and numerically unstable. In contrast, our method generates a sketch of an image in a single inference step after training, and is relatively more stable and suitable as a representation.

## 3. Mathematical framework

In this section, we present our theoretical analysis on how sketches can facilitate geometry-aware representation. We begin with a formal definition of sketch and strokes. We then introduce a mathematical framework based on geometric deep learning, which aims to impose the geometric prior on the learned feature.[1] Finally, we show that with a loss function that satisfies certain conditions, strokes can be equivariant with respect to arbitrary affine transformations and thus preserve geometric information.

### 3.1. Preliminaries and notations

We consider an image $I \in \mathcal{I}$ as *signals* on a physical domain $\Omega$, which maps pixel coordinates to the $C$ color channels, *i.e.*, $I : \Omega \to \mathbb{R}^C$. Although $\Omega$ is a 2D grid space for the real image, we assume $\Omega$ as a homogeneous space, where each border is connected to the opposite side.

**Stroke and sketch.** A ***stroke*** is a tuple $p = (t, c, w)$ composed of $k$ control points $t \in \Omega^k$, color $c \in [0, 1]^C$, and thickness $w \in [0, 1]$ of the Bézier curve. A rendering process $r : \mathcal{P} \to \mathcal{S}(\subset \mathcal{I})$ converts a set of $n$ strokes $\boldsymbol{p} = \{p^{(1)}, \cdots, p^{(n)}\} \in \mathcal{P}$ into the corresponding image (see Appendix D for further details), which we denote as ***sketch*** $S = r(\boldsymbol{p}) \in \mathcal{S}$. Let $f : \mathcal{I} \to \mathcal{P}$ be the embedding function which maps an image to a set of strokes. We denote $f(I) \in \mathcal{P}$ and $[r \circ f](I) \in \mathcal{S}$ respectively as a ***stroke representation*** and a ***sketch representation*** of the image $I$ w.r.t $f$. To generate a sketch that accurately reflects the given image, we adopt a metric function $\mathcal{L} : \mathcal{I} \times \mathcal{S} \to \mathbb{R}$ that measures the distance between the image and the sketch.

**Definition 1.** *We denote $S_I = \arg\min_{S \in \mathcal{S}} \mathcal{L}(I, S)$ as an* ***optimal sketch representation*** *of $I$.*

**Geometry-aware representation.** We denote $\mathcal{G}$ as a group of geometric transformations that acts on points in $\Omega$ (see Appendix E for definitions). For example, $\mathcal{G}$ could be a group of translation or rotation. $\rho$ is a group representation of $g \in \mathcal{G}$ such that $\rho(g)$ is a linear operator which acts on $\mathcal{I}$, *i.e.*, $[\rho(g)I](u) = I(g^{-1}u)$ for $u \in \Omega$.

**Definition 2.** *$f(I)$ is a* ***geometry-aware representation*** *of $I$ with respect to $\mathcal{G}$ if there exists a group representation $\rho'$ such that $f(\rho(g)I) = \rho'(g)f(I)$ for $\forall I \in \mathcal{I}$ and $\forall g \in \mathcal{G}$.*

In this case, we say $f$ is a $\mathcal{G}$**-equivariant map**, and the symmetry of $\mathcal{G}$ is preserved by $f$. For example, a convolution operator is a translation-equivariant map since the feature of the shifted image is equal to the shifted feature of

[1] See [3] for the extensive review.

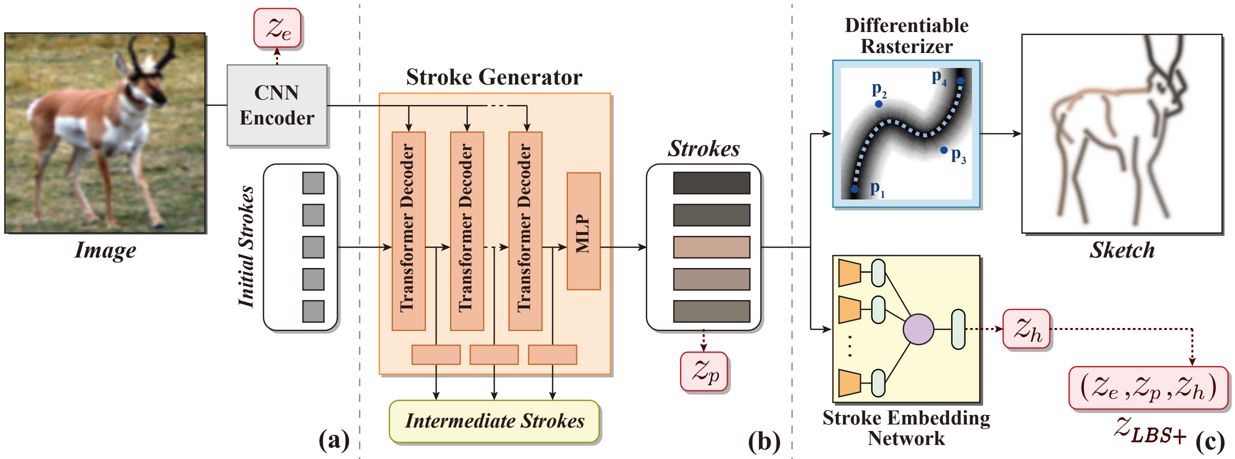


Figure 3. Visualization of the overall architecture of LBS. **(a)** A CNN-based encoder extracts the feature of the image. **(b)** Initial strokes are updated repeatedly with a stroke generator based on a Transformer decoder. For the progressive optimization process, intermediate strokes are extracted from each intermediate layer. **(c)** The rasterized sketch and stroke embedding $z_h$ are generated from the strokes. Features derived from the CNN encoder, strokes, and stroke embedding are combined to form the final representation, $z_{LBS+}$.

the original image. In other words, the convolution operator preserves the symmetry of the translation and produces a geometry-aware representation with respect to it.

### 3.2. Geometry-aware representations by sketching

We now show that stroke representation is a geometry-aware representation with respect to arbitrary affine transformations. Refer to Appendix E for detailed proofs.

**Proposition 1.** *If $\mathcal{L}(\rho(g)I, \rho(g)S) = \lambda(g)\mathcal{L}(I, S)$ where $\lambda : \mathcal{G} \to \mathbb{R}^+$ exists, and there exists a unique $f^*$ such that $[r \circ f^*](I) = S_I$ for $\forall I \in \mathcal{I}$, then $f^*(I)$ is a geometry-aware representation w.r.t. affine transformations $\mathcal{A}$.*

Prop. 1 states the conditions for $f$ to be an $\mathcal{A}$-equivariant map. Therefore, by designing stroke representations to be injective for $r$, and optimizing $f$ to minimize $\mathcal{L}$ subject to the conditions outlined in Prop. 1, $f$ generates a geometry-aware representation with respect to arbitrary affine transformations. In the next section, we propose an architecture and a loss function that satisfy these conditions.

## 4. Method

We describe our model LBS, which provides geometry-aware representations from strokes by learning to generate a sketch in a short inference time without using a sketch dataset. We introduce an objective function and techniques for training LBS. We then propose a stroke embedding network that aggregates information of whole strokes. Hereafter, we use the terms "sketch" and "image" to denote realized sketch $S(\Omega)$ and realized image $I(\Omega)$ for simplicity.

### 4.1. Learning by Sketching (LBS)

LBS aims to generate a set of strokes to sketch the image and extract geometric information with color for downstream tasks. To achieve the goal, we design a stroke generator by referring to Transformer-based sketch generation models [48, 60].

**Architecture.** As shown in Fig. 3, LBS generates a set of strokes from trainable parameters, which are denoted as initial strokes. Our stroke generator uses Transformer decoder layers and incorporates image features from a CNN encoder (ResNet18 is used in this paper). Then, 2-layer MLP decoders decode a set of $n$ strokes from the output of the Transformer decoder. Each stroke contains four control points that parameterize a cubic Bézier curve, as well as color and thickness information. The sketch of an image is then created by rasterizing the generated set of strokes using a differentiable rasterizer [53]. For a detailed description of our architecture, please refer to Appendix C.

**Representation of LBS.** The primary output of LBS is a set of strokes $\boldsymbol{p}$, and a geometry-aware representation is obtained by using the flattened vector of the strokes $z_p = (p^{(1)}, ..., p^{(n)})$. To complement information that may not be encoded with strokes such as texture, we concatenate $z_p$ with the image features $z_e$ from the CNN encoder with global average pooling. The resulting concatenated representation is denoted as $z_{LBS}$. In addition, we can combine this with representation $z_h$ that aggregates the entire stroke, the result of which we denote as $z_{LBS+}$. Further details can be found in Sec. 4.4. The final representation space of LBS is as follows:

$$z_{LBS} = (z_e, z_p), \quad z_{LBS+} = (z_e, z_p, z_h). \tag{1}$$

### 4.2. CLIP-based perceptual loss function

As discussed in Sec. 3.2, proper $\mathcal{L}$ must be given to train a geometry-aware representation of an image. To this end, we adopt CLIP-based perceptual loss to measure the sim-

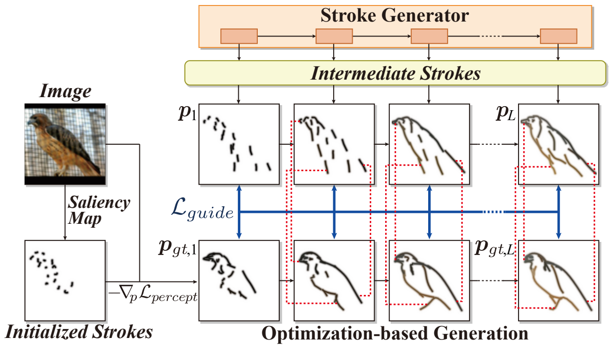


Figure 4. Visualization of the progressive optimization process. Strokes initialized from the saliency map are optimized using the gradient of $\mathcal{L}_{percept}$ until convergence and stored in a total of $L$ intermediate steps denoted by $\boldsymbol{p}_{gt,l}$. Then, the stroke generator is trained through $\mathcal{L}_{guide}$ to progressively predict each intermediate stroke $\boldsymbol{p}_l$ from $\boldsymbol{p}_{gt,l}$.

ilarity between the sketch and the input image in terms of both geometric and semantic differences [68]. To generate a sketch that is *semantically similar* to the input image, it should minimize the distance on the embedding space of the CLIP model $f_C$ as follows:

$$\mathcal{L}_{semantic} = \sum_{A \in \mathcal{A}} \phi(A(I), A(S)), \tag{2}$$

where $\mathcal{A}$ is the group of affine transformations and $\phi$ is the cosine similarity of the CLIP embeddings, *i.e.*, $\phi(x, y) = \cos(f_C(x), f_C(y))$. In addition, it compares low-level features of the sketch and the input image to estimate the *geometric similarity* as follows:

$$\mathcal{L}_{geometric} = \sum_{A \in \mathcal{A}} \sum_{i} \phi_i(A(I), A(S)), \tag{3}$$

where $\phi_i$ is the $\mathcal{L}_2$ distance of the embeddings from the $i$-th intermediate layer of the CLIP model, *i.e.*, $\phi_i(x, y) = \|f_{C,i}(x) - f_{C,i}(y)\|_2^2$. Specifically, we used $i \in \{3, 4\}$ with the ResNet101 CLIP model, proposed as [68]. The final perceptual loss is as follows:

$$\mathcal{L}_{percept} = \mathcal{L}_{geometric} + \lambda_s \cdot \mathcal{L}_{semantic}, \tag{4}$$

with $\lambda_s = 0.1$ as a hyperparameter.

### 4.3. Training with progressive optimization process

Abstracting an image into a sketch requires high-level reasoning, and predicting the corresponding stroke parameters in a single inference step is a delicate process. Although CLIPasso [68] addresses these problems via an optimization-based approach, its time-consuming properties (∼2 min per image) make it impractical as a representation. Moreover, as CLIPasso reported that the changes in initial strokes significantly impacted the final results, directly optimizing each stroke through the optimizer for thousands of steps leads to highly variable results and numerical instability.

To generate a stable representation within a single inference step, we modify the optimization-based approach of *CLIPasso* and utilize it as desirable guidance for the stroke generator, which we call the guidance stroke and denote as $\boldsymbol{p}_{gt}$. To ensure high-quality sketch generation, each intermediate layer of the stroke generator is guided by strokes obtained at the corresponding intermediate step of the optimization process instead of relying solely on the final outcome. This process, depicted in Fig. 4, involves progressively updating strokes from the previous layer, which can be modeled by residual blocks from the Transformer architecture. By decomposing the prediction of the entire optimization process into predicting changes in relatively short intervals for each intermediate layer, our approach leads to better-quality sketches.

However, the process through which CLIPasso initializes strokes via stochastic sampling from the saliency map results in unstable outcomes. To address this issue, we select initialized strokes deterministically by selecting the most salient regions from the saliency map in a greedy manner. The color of the stroke is determined during the initialization step by referencing the corresponding region in the original image. More details are described in Appendix A.

To ensure that the guidance loss is invariant to the permutation of the order of each stroke, we use the Hungarian algorithm [41] which is widely used for the assignment problem. The guidance loss is as follows:

$$\mathcal{L}_{guide} = \sum_{l=1}^{L} \min_{\sigma} \sum_{i=1}^{n} \mathcal{L}_1(p_{gt,l}^{(i)}, p_l^{(\sigma(i))}), \tag{5}$$

where $\sigma$ is a permutation of the stroke index, and $p_l^{(i)}$ is the $i$-th decoded stroke from the $l$-th intermediate layer with $L$ layers in total. $\mathcal{L}_1$ is the L1 distance, and $p_{gt,l}^{(i)}$ is the guidance stroke corresponding to $p_l^{(\sigma(i))}$.

### 4.4. Stroke embedding network

As strokes primarily express localized and fine-grained details, they can be less suitable for representing coarse-grained information. To address this, we offer an additional representation that combines full strokes through a stroke embedding network. Stroke embedding network $h : \mathcal{P} \rightarrow \mathcal{Z}_h$ maps the set of $n$ strokes into embedding space $\mathcal{Z}_h$. Since the rasterized sketch is invariant to the stroke indices (except for contact point between strokes), we followed the permutation invariant model from DeepSets [80] to generate stroke embedding. The choice of the loss function $\mathcal{L}_{embed}$ to train $z_h \in \mathcal{Z}_h$ is unconstrained (*e.g.*, Cross-Entropy or InfoNCE loss in our paper). During the training phase, the gradient from $h$ is also passed to $f$ to capture coarse-grained information.

The final objective is as follows:

$$\mathcal{L}_{LBS} = \mathcal{L}_{percept} + \lambda_g \cdot \mathcal{L}_{guide} + \lambda_e \cdot \mathcal{L}_{embed}, \quad (6)$$

where $\lambda_g$ and $\lambda_e$ are hyperparameters (for specific values, refer to Appendix C).

## 5. Experiments

### 5.1. Research questions and task designs

We investigate the effectiveness and generalizability of our approach by designing multiple research questions, as well as corresponding downstream tasks for each question. Detailed information about each experimental setup is described in Appendix B.

**RQ1. Effects of equivariance:** *What is the benefit of equivariant representation, and does LBS achieve it?*

From the definition of equivariance, we expect that an equivariant model with respect to geometric transformation would have generalization capability for augmented images with transformation. To verify this, we conduct experiments on the Rotated MNIST (rotMNIST) dataset [22], which consists of rotated images from the MNIST dataset. We rotate images by 30° and 45° for training and classify the digit of images rotated by 0° and 90° to evaluate whether LBS generalizes the information from the training domain to the test domain. To verify that our model satisfies equivariance outside the training domain, we classify the rotation and reflection between an original image and the same image rotated with an interval of 90° and horizontally flipped.

**RQ2. Effects on geometric primitives and concepts:** *How well does LBS represent geometric primitives and understand their conceptual relationships?*

To evaluate whether LBS is suitable for representing geometric primitives and concepts, we perform a classification task on the Geoclidean dataset [29]. The Geoclidean dataset consists of realized images from a concept of Euclidean geometry, *e.g.*, black parallel lines on a white background. Geoclidean is divided into two categories: Geoclidean-Elements and Geoclidean-Constraints, visualized in Fig. 7 on Appendix B. By providing a limited training set consisting of only 10 images per concept, we evaluate whether LBS can effectively learn the high-level relationships between each primitive and generalize them across different examples by classifying the concept of the test image.

**RQ3. Local geometric information and spatial reasoning:** *How effectively does LBS reflect local geometric information and extend it to spatial reasoning tasks?*

We validate whether LBS can reflect the local geometric information of each object in a synthetic photo dataset consisting of multiple objects, using the CLEVR dataset [34].

Table 1. Quantitative results of LBS and the baselines evaluated on the corresponding tasks on the rotMNIST and Geoclidean.

| Label | Method | rotMNIST (30°, 45°) 0°, 90° | rotMNIST (30°, 45°) Rotation | Geoclidean Constraints | Geoclidean Elements |
|---|---|---|---|---|---|
| ✓ | CE | 62.72±1.89 | 73.14±0.35 | 53.89±1.58 | 70.57±4.29 |
| ✓ | SupCon [38] | 65.25±1.05 | 73.37±0.20 | 42.41±3.16 | 55.83±4.28 |
| ✓ | LtD-diff [54] | 60.09±1.16 | 28.91±2.15 | 57.26±2.19 | 69.47±2.11 |
| ✓ | E(2)-CNN [73] | **85.12**±0.17 | **77.00**±0.39 | **71.03**±1.94 | 69.28±1.46 |
| ✓ | **LBS (CE)** | 65.00±3.32 | 75.81±0.30 | 50.01±1.58 | **81.06**±3.14 |
| ✗ | SimCLR [9] | **47.66**±0.41 | **72.96**±0.84 | 32.04±0.64 | 65.14±4.11 |
| ✗ | $\beta$-TCVAE [8] | 37.84±1.06 | 60.14±0.32 | 17.18±1.35 | 33.82±1.64 |
| ✗ | GeoSSL [55] | - | - | 18.66±3.33 | 33.47±2.80 |
| ✗ | HoG [14] | 25.42 | 64.94 | 23.82 | 52.05 |
| ✗ | **LBS** | 47.31±2.53 | 70.02±0.64 | **47.43**±1.34 | **81.34**±0.16 |

We train our model with very limited descriptions, where the label for the entire scene is provided as the rightmost object or without any descriptions at all. We validate the effectiveness of our representation by evaluating its ability to successfully classify attributes that are not provided as labels, such as determining the color of the leftmost object. Additionally, we test its ability to perform simple spatial reasoning, such as shifting the rightmost object and inferring the attribute of the current rightmost object.

**RQ4. Domain transfer:** *Can the geometric concepts of LBS trained in a specific domain be extended to other domains?*

To investigate whether the learned representation within a specific image domain provides meaningful geometric information across other domains, we evaluate the model by shifting the distribution from the STL-10 [11] dataset to CLEVR and vice versa. Evaluation in the STL-10 dataset is conducted through object classification.

**RQ5. Traditional sketch tasks:** *Does geometry-aware representation from LBS help in traditional FG-SBIR tasks?*

We assess the impact of geometry-aware representations acquired by LBS on a conventional sketch task by evaluating them with the fine-grained sketch-based image retrieval (FG-SBIR) task on the QMUL-ShoeV2 Dataset [78]. The aim of FG-SBIR is to match a given sketch of a shoe to its corresponding photo image. Since geometric information is essential for distinguishing each shoe, we investigate whether leveraging representations from LBS can improve performance on the FG-SBIR task.

### 5.2. Experimental setup

**Implementation details.** As samples from rotMNIST and Geoclidean can be directly represented with a few strokes, we replace $\mathcal{L}_{percept}$ and $\mathcal{L}_{guide}$ with $\mathcal{L}_1$ loss, which does not rely on a pre-trained CLIP model. For the STL-10 dataset, providing mask information for the background with U2-Net [57] improves the quality of generated sketches. For $\mathcal{L}_{embed}$, we use Cross-Entropy loss for supervised training which we denote as LBS (CE) in Tab. 1, 2 and

Table 2. Quantitative results on CLEVR dataset for labeled and unlabeled settings and for CLIP and HoG features. RC, LC, BC refer to inferring the color of the rightmost, leftmost, bottommost object, and Size, Shape, Material refer to inferring the size, shape, material of the rightmost object, respectively. Third refers to inferring the color of the third object from the right, and Shift refers to predicting the color of the rightmost object after shifting the initial rightmost object to the left by 0.15 times the image width.

| Label | Method | RC | LC | BC | Size | Shape | Material | Third | Shift |
|---|---|---|---|---|---|---|---|---|---|
| ✓ | CE | 98.71$\pm$0.10 | 76.72$\pm$0.86 | 76.02$\pm$0.90 | 92.51$\pm$0.40 | 49.97$\pm$0.29 | 64.38$\pm$1.90 | 40.66$\pm$0.20 | **62.06**$\pm$1.62 |
| | SupCon [38] | **98.75**$\pm$0.08 | 63.82$\pm$2.36 | 66.04$\pm$2.65 | 91.88$\pm$0.37 | 49.15$\pm$0.85 | 59.20$\pm$0.57 | 37.93$\pm$0.27 | 56.05$\pm$2.56 |
| | LtD-diff [54] | 62.29$\pm$0.48 | 14.01$\pm$0.42 | 15.84$\pm$0.43 | 63.98$\pm$3.38 | 43.96$\pm$3.05 | 54.47$\pm$0.45 | 16.47$\pm$0.59 | 17.21$\pm$0.29 |
| | E(2)-CNN [73] | 98.50$\pm$0.10 | 70.10$\pm$1.15 | 73.51$\pm$2.50 | 89.84$\pm$0.46 | 45.85$\pm$0.93 | 63.05$\pm$0.59 | **41.95**$\pm$0.18 | 59.29$\pm$0.91 |
| | **LBS (CE)** | 97.49$\pm$0.22 | **81.79**$\pm$0.27 | **84.09**$\pm$0.84 | **93.22**$\pm$0.29 | **70.03**$\pm$0.68 | **86.84**$\pm$0.54 | 38.23$\pm$0.25 | 51.56$\pm$0.16 |
| ✗ | SimCLR [9] | 60.61$\pm$1.24 | 63.89$\pm$2.35 | 63.77$\pm$2.29 | 83.35$\pm$0.60 | 41.95$\pm$0.33 | 51.66$\pm$0.24 | 33.42$\pm$0.55 | 43.05$\pm$0.55 |
| | E(2)-CNN [73] | 53.50$\pm$7.30 | 54.60$\pm$7.34 | 55.52$\pm$7.60 | 83.52$\pm$1.56 | 42.06$\pm$1.12 | 53.04$\pm$2.81 | 30.74$\pm$2.84 | 38.03$\pm$4.44 |
| | $\beta$-TCVAE [8] | 17.09$\pm$0.20 | 17.50$\pm$0.33 | 20.04$\pm$0.71 | 71.27$\pm$0.10 | 36.30$\pm$0.10 | 56.11$\pm$0.13 | 15.38$\pm$0.19 | 16.35$\pm$0.18 |
| | GeoSSL [55] | 20.16$\pm$0.63 | 20.54$\pm$0.87 | 21.61$\pm$0.67 | 73.79$\pm$0.78 | 44.08$\pm$1.10 | 54.54$\pm$2.94 | 15.39$\pm$0.16 | 16.94$\pm$0.34 |
| | DefGrid [21] | 73.81$\pm$0.91 | 73.96$\pm$1.02 | 73.38$\pm$0.80 | 81.50$\pm$0.22 | 46.34$\pm$0.77 | 68.65$\pm$1.17 | 24.90$\pm$0.27 | 36.28$\pm$0.13 |
| | **LBS** | **84.31**$\pm$0.08 | **80.47**$\pm$0.78 | **83.00**$\pm$0.39 | **92.66**$\pm$0.41 | **70.01**$\pm$0.53 | **85.52**$\pm$0.43 | **37.41**$\pm$0.29 | **49.32**$\pm$0.17 |
| | CLIP [59] | 37.39 | 39.1 | 54.98 | 77.51 | 66.91 | 72.66 | 34.75 | 34.80 |
| | HoG [14] | 56.83 | 49.69 | 58.69 | 81.73 | 61.14 | 68.38 | 24.28 | 33.25 |

Table 3. Quantitative results of models trained on STL-10 dataset and evaluated on CLEVR dataset.

| Dataset | Method | RC | LC | BC | Size | Shape | Material | Third | Shift |
|---|---|---|---|---|---|---|---|---|---|
| STL-10 → CLEVR | LtD [54] | 18.70$\pm$0.02 | 19.15$\pm$0.77 | 21.69$\pm$1.05 | 73.13$\pm$3.21 | 47.88$\pm$1.13 | 59.11$\pm$2.34 | 16.47$\pm$0.59 | 17.21$\pm$0.29 |
| | SimCLR [9] | 26.97$\pm$0.28 | 26.26$\pm$0.88 | 38.88$\pm$0.19 | 82.33$\pm$0.64 | 56.88$\pm$1.23 | 77.39$\pm$0.35 | 23.45$\pm$0.35 | 24.95$\pm$0.17 |
| | E(2)-CNN [73] | 26.82$\pm$2.23 | 25.40$\pm$1.20 | 39.30$\pm$2.80 | 79.48$\pm$0.71 | 53.54$\pm$0.84 | 72.18$\pm$0.34 | 23.35$\pm$1.01 | 24.80$\pm$1.08 |
| | $\beta$-TCVAE [8] | 17.04$\pm$0.76 | 18.46$\pm$2.36 | 18.88$\pm$1.14 | 71.32$\pm$0.67 | 37.14$\pm$0.12 | 56.81$\pm$0.26 | 15.63$\pm$1.35 | 16.43$\pm$0.38 |
| | GeoSSL [55] | 14.41$\pm$0.55 | 14.73$\pm$0.11 | 14.37$\pm$0.66 | 77.47$\pm$5.17 | 48.10$\pm$1.90 | 55.63$\pm$2.86 | 13.71$\pm$0.45 | 13.63$\pm$0.13 |
| | DefGrid [21] | **66.74**$\pm$1.10 | **68.60**$\pm$0.08 | **68.35**$\pm$0.82 | 78.97$\pm$0.29 | 40.92$\pm$0.44 | 62.98$\pm$1.31 | 23.48$\pm$0.49 | 33.07$\pm$0.52 |
| | **LBS** | 53.87$\pm$2.72 | 53.85$\pm$2.20 | 61.84$\pm$2.23 | **92.46**$\pm$0.62 | **59.52**$\pm$0.45 | **80.06**$\pm$1.29 | **30.75**$\pm$0.41 | **39.95**$\pm$1.15 |

Table 4. Quantitative results of models trained on CLEVR dataset and evaluated on STL-10 dataset classification.

| | Labeled | | Unlabeled | |
|---|---|---|---|---|
| Dataset | Method | Accuracy | Method | Accuracy |
| CLEVR → STL-10 | CE | 46.15$\pm$0.12 | SimCLR [9] | 41.68$\pm$0.05 |
| | SupCon [38] | 43.41$\pm$0.36 | $\beta$-TCVAE [8] | 27.35$\pm$0.38 |
| | LtD-diff [54] | 50.81$\pm$0.67 | GeoSSL [55] | 35.93$\pm$0.96 |
| | E(2)-CNN [73] | 45.19$\pm$0.84 | E(2)-CNN [73] | 38.50$\pm$0.49 |
| | **LBS (CE)** | **56.48**$\pm$0.89 | DefGrid [21] | 33.13$\pm$0.17 |
| | | | **LBS** | **55.35**$\pm$0.18 |

4. For the unlabeled setting, the model is trained without $\mathcal{L}_{embed}$ and is denoted LBS with no parentheses. For evaluation, we use $z_{LBS+}$ for LBS (CE), and $z_{LBS}$ for LBS. All evaluations are performed with the common protocol of linear classification. For more detailed information on our implementation, please refer to Appendix C.

**Baselines.** We compare our models with:

- Models which only use image encoder: trained with Cross-Entropy, SimCLR [9], or SupCon loss [38].
- Equivariance models: E(2)-Equivariant Steerable CNNs (E(2)-CNN) [73], and work by Novotny *et al.*, denoted as GeoSSL [55].
- Disentanglement model: $\beta$-TCVAE [8].
- Sketch model for communication between agents: Learning to Draw (LtD) [54]
- Grid-based superpixel model: DefGrid [21]
- Handcrafted feature descriptor: HoG [14].
- Pre-trained CLIP model used for training LBS.

### 5.3. Results and analysis

**Analysis on RQ1 and RQ2:** Tab. 1 shows the experimental results for **RQ1** and **RQ2**, which can be summarized as: **(a)** Providing a strong inductive bias for equivariance, as shown in the case of E(2)-CNN, leads to good performance on the tasks of **RQ1**. While LBS outperformed LtD-diff, which also utilizes a sketch-based approach, it requires additional validation and refinement. **(b)** For **RQ2**, LBS performs substantially better than the other baselines except for the supervised setting on Geoclidean-Constraints, showing that it successfully captures geometric concepts. In particular, Geoclidean-Elements perform better when trained without labels. **(c)** Disentanglement methods and handcrafted features offer less informative features for both tasks.

**Analysis on RQ3:** Tab. 2 summarizes the experiments conducted on the CLEVR dataset, demonstrating that LBS provides valuable information for tasks of **RQ3**. The key observations from Tab. 2 are: **(a)** While the baselines in the

Table 5. Top-1 and Top-10 accuracy on QMUL-Shoe-v2 dataset, using Triplet Loss for $\mathcal{L}_{embed}$ in LBS.

| Method | Rep | Top-1 | Top-10 |
|---|---|---|---|
| LBS (Triplet) | $z_{LBS+}$ | 40.8±4.19 | 88.3±1.76 |
| | $z_e$ | 30.8±3.89 | 77.8±3.89 |
| Triplet | $z_e$ | 32.5±1.73 | 80.0±4.36 |

supervised setting struggle to generalize on unlearned attributes, LBS (CE) can provide more general information. **(b)** The results of LBS show a larger performance gap in the unlabeled setting, with its performance being almost as good as LBS (CE). **(c)** Embeddings from pre-trained CLIP perform less effectively than unlabeled baselines, such as SimCLR, on tasks related to local geometry. Also, hand-crafted features yielded limited effectiveness. **(d)** Methods that focus on preserving global transformations and disentanglement are not effective in environments of **RQ3**.

**Analysis on RQ4:** Tabs. 3 and 4 summarize the results for shifting domains from STL-10 to CLEVR, and vice versa. As shown in Fig. 11 on Appendix I, LBS trained on STL-10 can successfully describe scenes from CLEVR, while LBS trained on CLEVR can abstractly depict natural images from STL-10, leading to substantially high classification accuracy. These results indicate that learning geometric information, such as position and curvature, by sketching salient parts of a scene successfully induces learning of general features across domains. Furthermore, LtD-diff, which also utilizes a sketch-based approach, demonstrated good performance in transferring CLEVR to STL-10.

**Analysis on RQ5:** Tab. 5 presents the results for the FG-SBIR task. Compared to the baseline trained without sketching, LBS can accurately predict the corresponding shoe during the test phase. The huge drop in performance upon removing strokes and stroke embedding from $z_{LBS+}$ indicates that the geometry-aware representation provided by LBS is also valuable for traditional sketch tasks.

### 5.4. Comparisons with stroke-based methods

Tab. 6 compares LBS to other stroke-based generation methods on STL-10 dataset, and Fig. 5 visualizes the results of each method. The evaluation is based solely on strokes, and LBSonly uses $z_p$ as a representation. LBS ($\mathcal{L}_{percept}$) achieved superior results compared to LBS ($\mathcal{L}_1$) and the Paint Transformer [48] based on $\mathcal{L}_1$ loss. We also compare our results to CLIPasso modified to use our deterministic initialization process, suggesting that strokes generated from CLIPasso may not be suitable as a representation. LtD-diff, which generates sketches to solve a communication game, often omits important geometric and color information, resulting in low performance.

Table 6. Comparison between stroke-based generation methods.

| | RC | BC | Size | Shape | Material |
|---|---|---|---|---|---|
| LtD-diff [54] | 18.70 | 21.69 | 73.13 | 47.88 | 59.11 |
| Paint [48] | 51.96 | 60.40 | 77.35 | 44.30 | 70.02 |
| CLIPasso [68] | 12.55 | 13.55 | 55.35 | 35.40 | 50.00 |
| LBS ($\mathcal{L}_1$) | 34.43 | 57.81 | 78.32 | 40.99 | 61.59 |
| **LBS ($\mathcal{L}_{percept}$)** | **55.15** | **60.63** | **90.14** | **51.23** | **76.16** |

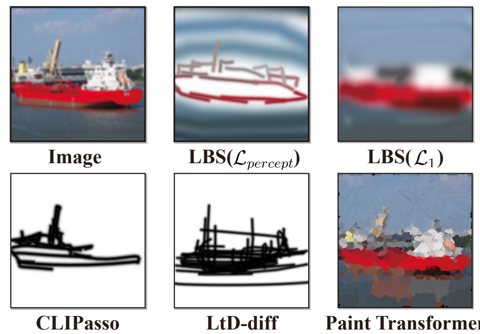


Figure 5. Comparisons to other stroke-based methods. LBS encodes both the geometric and semantic information.

## 6. Conclusion

We propose a novel representation learning perspective through sketching, a visual expression that retains geometric information. By considering the mathematical concept of equivariance, we provide a formal definition of sketch and stroke, and prove that strokes can preserve the information of arbitrary affine transformations. Our sketch generation model, LBS, minimizes CLIP-based perceptual loss and supports the idea that sketches can effectively convey geometric information. The experiments shed light on the potential benefits of leveraging the process of sketching to represent high-level geometric information explicitly. These benefits include learning equivariant representations, understanding geometric concepts, improving spatial reasoning abilities, and acquiring general geometric information across different domains.

Still, there are several limitations to our method. The training process of LBS relies on pre-trained CLIP and optionally U2-Net, and requires generating guidance strokes through a time-consuming optimization-based process. Moreover, while our work has demonstrated the theoretical strengths of sketching in terms of equivariance, a clearer methodology and further experimental analysis are needed to fully achieve this potential. Future work should address these issues and extend the methodology, focusing on scalability and including more complex and realistic tasks.

**Acknowledgement** This work was partly supported by the Korean government (2021-0-02068-AIHub/25%, 2021-0-01343-GSAI/25%, 2022-0-00951-LBA/25%, 2022-0-00953-PICA/25%).

## References


[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011. 3

[2] Hilton Bristow, Jack Valmadre, and Simon Lucey. Dense semantic correspondence where every pixel is a classifier. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4024–4031, 2015. 1, 2

[3] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021. 2, 3

[4] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986. 1, 2

[5] Nan Cao, Xin Yan, Yang Shi, and Chaoran Chen. Ai-sketcher: a deep generative model for producing high-quality sketches. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 2564–2571, 2019. 3

[6] Caroline Chan, Frédo Durand, and Phillip Isola. Learning to generate line drawings that convey geometry and semantics. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7915–7925, 2022. 3

[7] Salman Cheema, Sumit Gulwani, and Joseph LaViola. Quickdraw: Improving drawing experience for geometric diagrams. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, page 1037–1064, New York, NY, USA, 2012. Association for Computing Machinery. 3

[8] Ricky TQ Chen, Xuechen Li, Roger B Grosse, and David K Duvenaud. Isolating sources of disentanglement in variational autoencoders. *Advances in neural information processing systems*, 31, 2018. 1, 3, 6, 7, 16

[9] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR, 13–18 Jul 2020. 6, 7

[10] Yajing Chen, Shikui Tu, Yuqi Yi, and Lei Xu. Sketch-pix2seq: a model to generate sketches of multiple categories. *arXiv preprint arXiv:1709.04121*, 2017. 3

[11] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 6

[12] Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR. 1, 2

[13] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016. 1, 2, 16

[14] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 1, 2, 6, 7

[15] Fei Deng, Zhuo Zhi, Donghun Lee, and Sungjin Ahn. Generative scene graph networks. In *International Conference on Learning Representations*, 2021. 2

[16] Sounak Dey, Pau Riba, Anjan Dutta, Josep Llados, and Yi-Zhe Song. Doodle to search: Practical zero-shot sketch-based image retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2179–2188, 2019. 3

[17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 20

[18] Kevin Frans, Lisa B. Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, abs/2106.14843, 2021. 13

[19] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1652–1661. PMLR, 2018. 3

[20] Chengying Gao, Qi Liu, Qi Xu, Limin Wang, Jianzhuang Liu, and Changqing Zou. Sketchycoco: Image generation from freehand scene sketches. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5173–5182, 2020. 3

[21] Jun Gao, Zian Wang, Jinchen Xuan, and Sanja Fidler. Beyond fixed grid: Learning geometric image representation with a deformable grid. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part IX 16*, pages 108–125. Springer, 2020. 1, 2, 7, 16

[22] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, and David Balduzzi. Domain generalization for object recognition with multi-task autoencoders. In *Proceedings of the IEEE international conference on computer vision*, pages 2551–2559, 2015. 6

[23] Rohan Ghosh and Anupam K. Gupta. Scale steerable filters for locally scale-invariant convolutional neural networks. *CoRR*, abs/1906.03861, 2019. 1, 2

[24] David Ha and Douglas Eck. A neural representation of sketch drawings. In *International Conference on Learning Representations*, 2018. 3

[25] Xiaoguang Han, Chang Gao, and Yizhou Yu. Deepsketch2face: a deep learning based sketching system for 3d face and caricature modeling. *ACM Transactions on graphics (TOG)*, 36(4):1–12, 2017. 3

[26] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual repre-

sentation learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9726–9735, 2020. 15

[27] Irina Higgins, David Amos, David Pfau, Sébastien Racanière, Loïc Matthey, Danilo J. Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *CoRR*, abs/1812.02230, 2018. 1

[28] Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. beta-VAE: Learning basic visual concepts with a constrained variational framework. In *International Conference on Learning Representations*, 2017. 1, 3

[29] Joy Hsu, Jiajun Wu, and Noah Goodman. Geoclidean: Few-shot generalization in euclidean geometry. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 6

[30] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8709–8718, 2019. 3

[31] Junhwa Hur, Hwasup Lim, Changsoo Park, and Sang Chul Ahn. Generalized deformable spatial pyramid: Geometry-preserving dense correspondence estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1392–1400, 2015. 1, 2

[32] Varun Jampani, Deqing Sun, Ming-Yu Liu, Ming-Hsuan Yang, and Jan Kautz. Superpixel sampling networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 352–368, 2018. 2

[33] Youngjoo Jo and Jongyoul Park. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019. 3

[34] Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2901–2910, 2017. 6

[35] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker. Design of an image edge detection filter using the sobel operator. *IEEE Journal of solid-state circuits*, 23(2):358–367, 1988. 2

[36] Altaf Khan, Alexander Chefranov, and Hasan Demirel. Image scene geometry recognition using low-level features fusion at multi-layer deep cnn. *Neurocomputing*, 440:111–126, 2021. 1, 2

[37] Naman Khetan, Tushar Arora, Samee Ur Rehman, and Deepak K Gupta. Implicit equivariance in convolutional networks. *arXiv preprint arXiv:2111.14157*, 2021. 3

[38] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, 2020. 6, 7

[39] Hyunjik Kim and Andriy Mnih. Disentangling by factorising. In *International Conference on Machine Learning*, pages 2649–2658. PMLR, 2018. 1, 3

[40] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. 3

[41] Harold W. Kuhn. The Hungarian Method for the Assignment Problem. *Naval Research Logistics Quarterly*, 2(1–2):83–97, March 1955. 5

[42] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Bendsketch: Modeling freeform surfaces through 2d sketching. *ACM Transactions on Graphics (TOG)*, 36(4):1–14, 2017. 3

[43] Minchen Li. *Foldsketch: Enriching garments with physically reproducible folds*. PhD thesis, University of British Columbia, 2018. 3

[44] Mengtian Li, Zhe Lin, Radomir Mech, Ersin Yumer, and Deva Ramanan. Photo-sketching: Inferring contour drawings from images. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1403–1412. IEEE, 2019. 3

[45] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018. 3

[46] H. Lin, Y. Fu, X. Xue, and Y. Jiang. Sketch-bert: Learning sketch bidirectional encoder representation from transformers by self-supervised learning of sketch gestalt. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6757–6766, Los Alamitos, CA, USA, jun 2020. IEEE Computer Society. 3

[47] Li Liu, Fumin Shen, Yuming Shen, Xianglong Liu, and Ling Shao. Deep sketch hashing: Fast free-hand sketch-based image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2862–2871, 2017. 3

[48] Songhua Liu, Tianwei Lin, Dongliang He, Fu Li, Ruifeng Deng, Xin Li, Errui Ding, and Hao Wang. Paint transformer: Feed forward neural painting with stroke prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 3, 4, 8

[49] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[50] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 15

[51] Emile Mathieu, Tom Rainforth, Nana Siddharth, and Yee Whye Teh. Disentangling disentanglement in variational autoencoders. In *International Conference on Machine Learning*, pages 4402–4412. PMLR, 2019. 1, 3

[52] John F. J. Mellor, Eunbyung Park, Yaroslav Ganin, Igor Babuschkin, Tejas Kulkarni, Dan Rosenbaum, Andy Ballard, Theophane Weber, Oriol Vinyals, and S. M. Ali Eslami. Unsupervised doodling and painting with improved SPIRAL. *CoRR*, abs/1910.01007, 2019. 3

[53] Daniela Mihai and Jonathon Hare. Differentiable drawing and sketching. *arXiv preprint arXiv:2103.16194*, 2021. 3, 4, 16

[54] Daniela Mihai and Jonathon Hare. Learning to draw: Emergent communication through sketching. *Advances in Neural Information Processing Systems*, 34:7153–7166, 2021. 3, 6, 7, 8, 16

[55] David Novotny, Samuel Albanie, Diane Larlus, and Andrea Vedaldi. Self-supervised learning of geometrically stable features through probabilistic introspection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3637–3645, 2018. 1, 6, 7, 16

[56] Tiziano Portenier, Qiyang Hu, Attila Szabo, Siavash Bigdeli, Paolo Favaro, and Matthias Zwicker. Faceshop: Deep sketch-based face image editing. *ACM Transactions on Graphics*, 37, 04 2018. 3

[57] Xuebin Qin, Zichen Zhang, Chenyang Huang, Masood Dehghan, Osmar R Zaiane, and Martin Jagersand. U2-net: Going deeper with nested u-structure for salient object detection. *Pattern recognition*, 106:107404, 2020. 1, 2, 6, 13

[58] Robin Quessard, Thomas Barrett, and William Clements. Learning disentangled representations and group structure of dynamical environments. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19727–19737. Curran Associates, Inc., 2020. 1, 3

[59] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR, 18–24 Jul 2021. 2, 3, 7

[60] Leo Sampaio Ferraz Ribeiro, Tu Bui, John Collomosse, and Moacir Ponti. Sketchformer: Transformer-based representation for sketched structure. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14153–14162, 2020. 3, 4

[61] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[62] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, 2011. 2

[63] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: Learning to retrieve badly drawn bunnies. *ACM Trans. Graph.*, 35(4), jul 2016. 3

[64] Lukas Schott, Julius von Kugelgen, Frederik Traeuble, Peter Gehler, Chris Russell, Matthias Bethge, Bernhard Schölkopf, Francesco Locatello, and Wieland Brendel. Visual representation learning does not generalize strongly within the same domain. In *ICLR 2022*, 2022. 1, 3

[65] Jiaxin Shi, Hanwang Zhang, and Juanzi Li. Explainable and explicit visual reasoning over scene graphs. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8376–8384, 2019. 2

[66] Nathalie Sinclair, Joan Moss, Zachary Hawes, and Carol Stephenson. *Learning Through and from Drawing in Early Years Geometry*, pages 229–252. Springer International Publishing, Cham, 2018. 2

[67] Jifei Song, Kaiyue Pang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning to sketch with shortcut cycle consistency. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 801–810, 2018. 3

[68] Yael Vinker, Ehsan Pajouheshgar, Jessica Y. Bo, Roman Christian Bachmann, Amit Haim Bermano, Daniel Cohen-Or, Amir Zamir, and Ariel Shamir. Clipasso: semantically-aware object sketching. *ACM Trans. Graph.*, 41(4):86:1–86:11, 2022. 2, 3, 5, 8, 13

[69] Sheng-Yu Wang, David Bau, and Jun-Yan Zhu. Sketch your own gan. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 14050–14060, 2021. 3

[70] Tan Wang, Zhongqi Yue, Jianqiang Huang, Qianru Sun, and Hanwang Zhang. Self-supervised learning disentangled group representation as feature. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. 3

[71] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovic, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. *ACM Trans. Graph.*, 37(6):1:1–1:14, 2018. 3

[72] Xiaogang Wang and Xiaoou Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1955–1967, 2009. 3

[73] Maurice Weiler and Gabriele Cesa. General e (2)-equivariant steerable cnns. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2, 6, 7

[74] Holger Winnemöller, Jan Eric Kyprianidis, and Sven C Olsen. Xdog: An extended difference-of-gaussians compendium including advanced image stylization. *Computers & Graphics*, 36(6):740–753, 2012. 2, 13

[75] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015. 2

[76] Peipei Yang, Kaizhu Huang, and Cheng-Lin Liu. Geometry preserving multi-task metric learning. *Machine learning*, 92(1):133–175, 2013. 1

[77] Ran Yi, Yong-Jin Liu, Yu-Kun Lai, and Paul L Rosin. Apdrawinggan: Generating artistic portrait drawings from face photos with hierarchical gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10743–10752, 2019. 3

[78] Qian Yu, Feng Liu, Yi-Zhe Song, Tao Xiang, Timothy M Hospedales, and Chen-Change Loy. Sketch me that shoe. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 799–807, 2016. 3, 6

[79] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 16

[80] Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017. 5, 15

[81] Ningyuan Zheng, Yifan Jiang, and Dingjiang Huang. Strokenet: A neural painting environment. In *International Conference on Learning Representations*, 2018. 3

[82] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 3

[83] Zhengxia Zou, Tianyang Shi, Shuang Qiu, Yi Yuan, and Zhenwei Shi. Stylized neural painting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15689–15698, 2021. 3