Editable Image Geometric Abstraction via Neural Primitive Assembly

Ye Chen[1] Bingbing Ni[1,2*] Xuanhong Chen[1,2] Zhangli Hu[1]
[1]Shanghai Jiao Tong University, Shanghai 200240, China
[2]USC-SJTU Institute of Cultural and Creative Industry
{chenye123, nibingbing, chen19910528, tension-2019}@sjtu.edu.cn

Abstract

This work explores a novel image geometric abstraction paradigm based on assembly out of a pool of predefined simple parametric primitives (i.e., triangle, rectangle, circle and semicircle), facilitating controllable shape editing in images. While cast as a mixed combinatorial and continuous optimization problem, the above task is approximately reformulated within a token translation neural framework that simultaneously outputs primitive assignments and corresponding transformation and color parameters in an image-to-set manner, thus bypassing complex/non-differentiable graph-matching iterations. To relax the searching space and address the vanishing gradient issue, a novel Neural Soft Assignment scheme that well explores the quasi-equivalence between the assignment in Bipartite b-Matching and opacity-aware weighted multiple rasterization combination is introduced, drastically reducing the optimization complexity. Without ground-truth image abstraction labeling (i.e., vectorized representation), the whole pipeline is end-to-end trainable in a self-supervised manner, based on the linkage of differentiable rasterization techniques. Extensive experiments on several datasets well demonstrate that our framework is able to predict highly compelling vectorized geometric abstraction results with a combination of ONLY four simple primitives, also with VERY straightforward shape editing capability by simple replacement of primitive type, compared to previous image abstraction and image vectorization methods.

1. Introduction

Image abstraction, as one of the fundamental tasks in computer vision, originally aims at decomposing a given image into a set of elementary primitives (*e.g.*, basic shapes, curves) as well as their inter-relationship, for the purpose of visual understanding or object reconstruction [6, 14, 13]. Haeberli *et al*. [16] use texture maps as primitives for painting and anti-aliased text drawing; while Hu *et al*. [19] adopt statistical learning strategy for approximating Ghost Imaging (GI [30]) by a set of natural image patches. Contemporary abstraction approaches [24, 41, 20, 28] delve to represent image with parameterized sketches. These methods usually represent sketches based on Bézier control points, and the input image is *translated* into a sequence of Bézier curves in a recursive way, based on deep architectures such as transformers [34] or reinforcement learning [28].

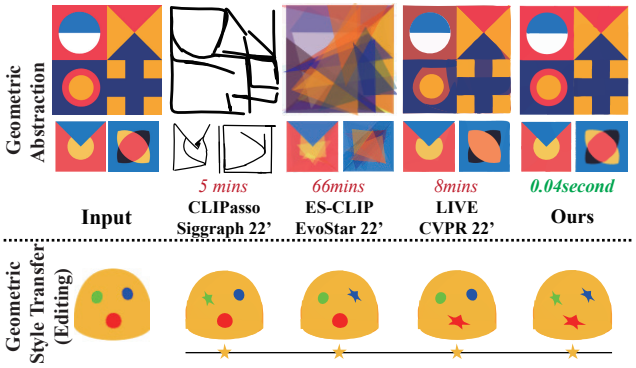

Figure 1. This work explores abstracting images based on a combination of simple geometric primitives (*i.e.*, geometric abstraction). Our method predicts highly compelling vectorized results to characterize the image geometric information efficiently. In addition, our method has straightforward shape editing capability by simply modifying primitives during inference.

Other than sketches, geometric shape primitive (triangle, circle, *etc*.) based image abstraction approaches [9, 39] are probably more favored by highly frequent applications such as graphic design, industrial design, cartoon production, advertisement making and modern arts, given the following facts. 1. **Simplicity and Expressibility**: On one hand, geometric primitive has very compact and explicit parameter format, thus facilitating compact representation and convenient manipulation. On the other hand, numerous combinations of geometric primitives imply rich expressive power compared with sketches. For example, the Bauhaus and Cubism arts tend to adopt ONLY simplistic shapes like rectangles and spheres, yet still with extremely rich artistic expression. 2. **Semantic Preservation**: Human beings tend to

*Corresponding author: Bingbing Ni

think of complex visual patterns abstracted into some primitives and their combinations. For example, a big circle, two small circles and two triangles sufficiently form a human face. These are two of the main reasons why modern logo design and animation especially favor the use of geometric primitives. This work is dedicated to geometric primitive learning for reassembling and editing artistic images.

Unfortunately, searching (or even approximating) the optimal solution of the above problem is considered as *hard* as it essentially invokes both combinatorial (one-to-many matching for best primitive type selection) and continuous (inference of the best transformation parameter for a certain primitive) simultaneously. The following challenges have to be addressed. 1.**High Dimensionality**: Compared to sketch-based methods that only need to predict control point locations, geometric abstraction requires selection of primitive types, leading to exponentially exploding combinatorial space. In addition, more complicated transformation parameters such as affine coefficients and colors need to be predicted simultaneously, which further expands the optimization space. 2.**Non-Differentiability**: Note that the non-differentiable nature of primitive selection together with primitive-to-pixel rasterization double the difficulty in employing any off-the-shelf deep learning framework. More seriously, the enclosed regions within primitives are usually flat and textureless, therefore popular differentiable rasterization techniques such as Diffvg [23] can hardly capture accurate gradient signal propagated backward for primitive type or transform parameter correction, *i.e*., known as vanishing gradient [18]. This issue makes training process highly unstable and difficult to generate meaningful abstract results. 3.**Weak Supervision**: Although steady progress in un-/self-supervised representation learning has emerged with encouraging results on multiple 2D/3D visual tasks [17, 8, 47, 26, 25], existing image abstraction methods [3, 28, 31] mostly rely on abundant datasets (*e.g*., Quickdraw [15]) and carefully designed loss metrics [41, 11], where vectorized sketch labeling is available for direct/strong pair-wise supervision. However, in this work we ONLY have rasterized images for training without any image-to-vector or image-to-primitive labeling, therefore the primitive assignment task is in an unsupervised (or self-supervised) setting since large-scale high-quality artistic images in parametric/vectorized format are extremely difficult to collect.

In pursuit of geometric abstraction and addressing the aforementioned difficulties, we propose a novel image geometric abstraction paradigm based on assembly out of a pool of pre-defined simple parametric primitives (triangle, rectangle, circle and semicircle), facilitating controllable shape editing in images. Our framework features the following designs. Architecturally, the above task is approximately reformulated within a token translation neural framework (*i.e*., transformer architecture [40]) that simultaneously outputs primitive assignments and the corresponding transformation parameters and colors in an image-to-set manner, thus bypassing complex/non-differentiable primitive assignment iterations. To avoid exhaustive combinatorial search, we introduce a novel Neural Soft Assignment scheme that well explores the quasi-equivalence between the selection operation in Bipartite b-Matching and opacity-aware weighted multiple rasterization combination. Namely, for each considered image location, a single rendering pattern is obtained by firstly fusing multiple primitive selections (with estimated transformations) and then softly inferring primitive matching weights serving as *opacity* values, in favor of differentiable learning of primitive assignment. In addition, to address vanishing gradient challenge, we propose a Centroid Filtering mechanism to calibrate gradients inside enclosed regions, in heart of which is a centroid smoothing kernel that creates an effective gradient propagation path for flat primitives. Without ground-truth image abstraction labeling (*i.e*., vectorized representation), the whole pipeline is end-to-end trainable in an unsupervised manner, based on the linkage of a differentiable rasterization technique [23].

We extensively experiment with the proposed framework both qualitatively and quantitatively on various datasets including Emoji [2], Icon [1] and Bauhaus-style graphic designs. It is demonstrated that our method is able to generate highly compelling results given only a small number of simple geometric shape primitive prototypes. Compared with sketch-based methods that easily generate redundant and mismatched curves, our method realizes compact combination of several primitives with good approximation accuracy. Moreover, our method can generate novel vectorized shapes with rich semantics by simply replacing/editing the primitive type during inference, which also indicates its great applicability for modern graphic design.

2. Related Works

Image Abstraction. Contemporary image abstraction techniques mainly focus on representing images using sketches due to their simplicity [36, 22, 31]. Recent efforts [11, 41, 39] employ CLIP [32] model to facilitate image abstraction because of its remarkable ability to distill semantic concepts from sketches and images alike. For example, CLIPasso [41] utilizes a pretrained CLIP model to provide geometric and semantic guidance and generates sketches for input images with multiple levels of abstraction. Besides sketch-based image abstraction, **geometric abstraction** [9] is another influential image abstraction paradigm. Tian *et al*. [39] combine CLIP model with evolution strategies [5] and abstract images by optimizing the procedural placement of parametric triangles. However, it is obviously not enough to abstract images with varying structures effectively with

only triangles. Also, results in [39] are limited to either fully semantic (CLIP guided) or color mean error (MSE guided). In our work, we explore geometric abstraction with a primitive set containing diverse geometric elements like rectangle and circle, *etc*. Our task requires additional selection of primitives, which is non-differentiable and increases the difficulty of optimization.

Image Vectorization. Our method represents input images with vector graphic primitives. Previous works [10, 44, 46] in this area focus on pre-segmenting images and then regressing segmented components to vector graphics. In the era of deep learning, researchers perform image vectorization with differentiable rendering/rasterization [23]. Im2Vec [33] proposes an RNN-based VAE model to predict vector paths for raster images without vector supervision. LIVE [27] emphasizes the importance of path initialization for the image vectorization task, which proposes component-wise path initialization and hierarchically optimizes the vector graphics in a layer-wise manner. Despite achieving compelling vectorization results, LIVE is inefficient because of its single-pass optimization. Our method formulates image vectorization as an image-to-set problem and synthesizes high-quality/regular vector graphics through combination of simple parametric primitives efficiently.

Reconstruction with primitives. Our work is also related to primitive-based reconstruction, which is commonly used in CAD designs [12, 35, 29, 48] and image/sketch compression [3, 43, 45, 38]. One closely related work is PMN [3], which abstracts sketches by matching them with pre-defined primitives in a supervised manner, benefiting from large-scale sketch datasets. Different from aforementioned methods, our work proposes to reconstruct raster images with pre-defined parametric primitives without vector graphics supervision, which is a much harder task.

3. Methodology

3.1. Problem Definition

Given an input raster image $\boldsymbol{I}$, our task is to reassemble it with parametric primitives. To this end, we firstly define a set of primitives $\mathbb{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_n\}$ containing n geometric shape types such as triangle, rectangle and circle, where each primitive is defined as a sequence of t points. To sample and aggregate the primitives to represent the input image, we train a model to decode the input image to a transformation set $\mathbb{T} = \{\mathcal{T}_1, ..., \mathcal{T}_m\}$ with each element including affine transformation coefficients and m denotes the number of parametric shapes needed to form the image. For each transformation element, we choose its primitive type and transform the primitive with corresponding affine parameters. Note that different elements can select the same primitive type. Hence we formulate the task as a Bipartite b-Matching problem between the transformation set and primitive set with $b = m$ and our task is to yield a primitive assignment matrix $\boldsymbol{A} = [a_{ij}]_{m \times n}$, where $a_{ij} \in \{0, 1\}$ and $a_{ij} = 1$ means an assignment $(\mathcal{T}_i, \boldsymbol{p}_j)$ exists. Then we can obtain a set of transformed shapes $\mathbb{S} = \{\boldsymbol{s}_i = \mathcal{T}_i(\boldsymbol{p}_j) | 1 \leq i \leq m, 1 \leq j \leq n, \forall a_{ij} = 1\}$ to be rasterized as the geometric abstraction result and the optimization problem is defined as:

$$\begin{aligned} &\min_{\boldsymbol{A}=[a_{ij}]_{m\times n}} \mathcal{L}(\boldsymbol{I}, \mathcal{R}(\mathbb{S})), \\ &s.t. \quad a_{ij} \in \{0, 1\}, 1 \leq i \leq m, 1 \leq j \leq n, \\ &\quad \sum_j a_{ij} = 1, i = 1, 2, 3, ..., m, \end{aligned} \tag{1}$$

where $\mathcal{R}$ denotes the rasterization function and $\mathcal{L}$ is the loss function evaluating pixel errors. This task is difficult to optimize with commonly used Graph Matching solvers like Hungarian algorithm [21] or reinforcement learning based methods [42, 4] because of the high dimensionality, non-differentiability and weak supervision as stated in Sec. 1.

For the convenience of reading, we collectively use *primitive* to represent the pre-defined geometric primitives (*i.e*., elements in $\mathbb{P}$) and use *shape* to denote the transformed primitive instances with corresponding transformation parameters and colors for rasterization (*i.e*., elements in $\mathbb{S}$).

3.2. Shape Attribute Set Prediction

To sample a certain number of primitives and transform them to shapes for rasterization, we train the model to predict a shape attribute set, where each element contains primitive type, transformation parameters and colors.

We firstly employ an image auto-encoder to embed the input image into hierarchical features/image tokens (including the high-level feature ($\boldsymbol{H}_t$) containing rich global information and the low-level feature ($\boldsymbol{H}_b$) representing local details) and decode the features to reconstruct input image, which provides effective guidance for shape attribute set prediction.

We abandon the usual recursive strategy in the stroke generation problem (*i.e*., the parameter generation of the next stroke depends on the output of the previous stroke and the input of the current feature) because this serial strategy not only requires iteration but also is prone to error accumulation. In particular, this method cannot explicitly explore the layout and structural relationship of various elements in the picture at medium and long spatial distances.

For our problem, *i.e*., image analysis and reconstruction based on primitives, it is very critical to infer which primitive (as well as its transformation and color parameters) should fill a certain image position/region through spatial context. Based on the above considerations, we propose to use the transformer framework [40] to output in parallel

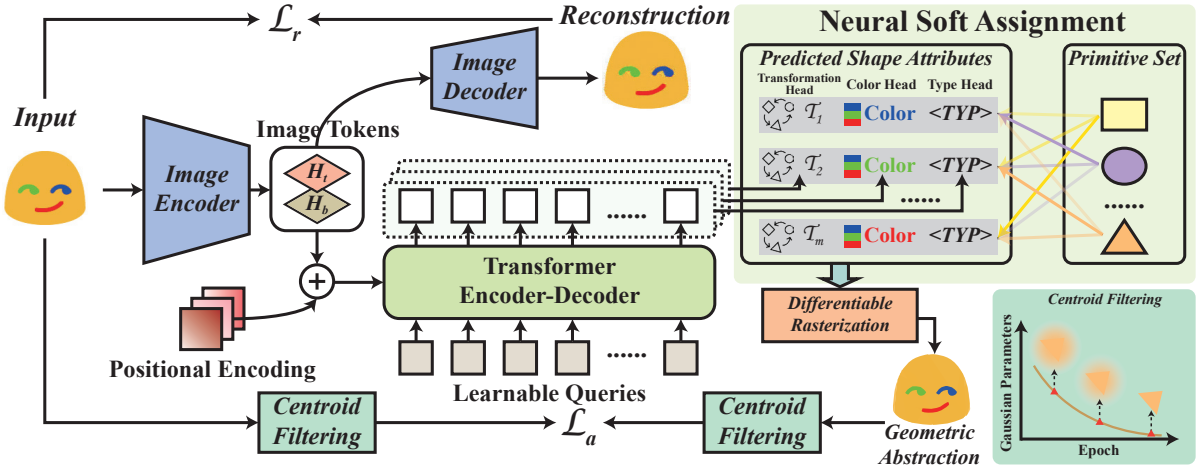

Figure 2. **Overview of our framework.** Our framework firstly uses an image auto-encoder to map the input image to image tokens, which are fed into an image decoder to reconstruct the input and a transformer-based network to predict a set of shape attributes (*i.e.*, transformation parameters, colors and primitive type classification probabilities) by three heads. Then the shape attribute set is matched with the pre-defined parametric primitive set through our neural soft assignment scheme to generate vector graphics to abstract the input image geometrically in a self-supervised manner with the help of differentiable rasterization [23].

all shape attributes (like affine coefficients and colors) used to match with primitives and form the whole input image. In other words, the goal is to sample a certain number of primitives and place them at specific image positions with appropriate geometric and appearance attributes in parallel. The advantages of this method are two-fold. On one hand, the image can be divided into a finite number of regions through the position encoding mechanism to guide the local best matching (avoiding exhaustive global search). On the other hand, the global inter-dependency among shapes can be fully modeled through the token interaction mechanism. Our shape attribute set prediction follows the architecture of DETR [7], a classic feed-forward set prediction network. Specifically, the image features extracted by the image encoder concatenated with learnable positional encodings are used as the input for the transformer encoder. Then the transformer decoder takes m learnable queries as input and transforms them into output embeddings,*i.e.*, $z \in \mathbb{R}^{m\times d}$. The output embeddings are then in parallel decoded by three prediction heads, *i.e.*, a primitive type prediction head, a transformation prediction head and a color prediction head. **Primitive type prediction**: For the primitive set defined in Sec. 3.1, each transformation (out of m) has n possible choices. Hence the primitive type prediction head maps the embedding into a n-dimensional probability distribution by softmax, *i.e.*, $\mathcal{F}_w : z \mapsto \boldsymbol{W} \in \mathbb{R}^{m\times n}$, which relaxes $\boldsymbol{A}$ to a probability distribution. **Transformation prediction**: To well reconstruct/approximate various complex shapes, we define an affine transformation matrix $\boldsymbol{M}$ to transform primitives in the order of *rotation-scale-translation-shear* with rotation factor α, scale factors (s_x, s_y), translation factors (t_x, t_y) and shear factors (β_x, β_y). Therefore, the affine transformation prediction can be defined as the mapping: $\mathcal{F}_s : z \mapsto \boldsymbol{\theta} \in \mathbb{R}^{m\times 7}$. To make the transformed primitives fit the shapes more precisely, we equip the affine transformation with point-wise offsets $\boldsymbol{\Delta} \in \mathbb{R}^{m\times 2t}$ as refinement, where t is the number of parametric points of each primitive. Hence, each transformation $\mathcal{T}_i$ can be regarded as a combination of $\boldsymbol{M}(\boldsymbol{\theta}_i)$ and $\boldsymbol{\Delta}_i$. **Color prediction**: Also, for each embedding we also map it to colors: $\mathcal{F}_c : z \mapsto \boldsymbol{C} \in \mathbb{R}^{m\times 3}$.

Note that during inference, we directly select the primitive shape type with the highest probability and raster it according to the corresponding transformation and color.

3.3. Bipartite b-Matching with Neural Soft Assignment

Pursuit of the optimal primitive selection is a Bipartite b-Matching problem. Commonly, the Hungarian algorithm [21] could be utilized [7, 37], with iterative adding or removing matching edges. However, the Hungarian algorithm is NOT applicable in our task since the vector graphics supervision (we only have pixel images for training) for the input is NOT available and thus we cannot make pairwise comparisons between each assignment [37]. More seriously, for each matching candidate, we need to perform a separate rasterization for pixel-level loss evaluation. For a transformation set containing m elements and a primitive set with n shape types, the number of rasterization times is n^m, which is computationally infeasible.

We note that for each image location, previous algorithms might select certain primitives with low probability. The pixel patterns formed by rasterization of these low-probability primitives and the ground-truth image have large discrepancies, so the gradients introduced by them are relatively unreliable. We also note that in the standard

rasterization algorithm, different graphic primitives/shapes at the same position can be superimposed and fused with the value of transparency, which means that those shapes with high transparency have less contribution towards the final rendered pattern. Namely, in differentiable rasterization methods, opacity values are always defined to describe the degree to which content behind an element/shape is hidden. Inspired by this observation, we propose a relaxed Bipartite b-Matching objective. The core idea is that the output probability of all possible primitives at each position could serve as the opacity value so that all possible candidates can be combined and superimposed and then rasterized together just for once. In this way, we can greatly reduce the number of rasterization times for different candidates during optimization, and make it easier to back-propagate the gradient towards the primitive closest to the accurate shape.

Specifically, for a shape $\boldsymbol{s}_i$ with color $\boldsymbol{c}_i$ and opacity o_i, the amount of appearance it contributes to the rasterized image can be regarded as $o_i\boldsymbol{c}_i$ and the rasterized image $\boldsymbol{I}^{\mathcal{R}}$ can be described as:

$$\boldsymbol{I}^{\mathcal{R}} = \mathcal{R}\big(\{(\boldsymbol{s}_1, o_1\boldsymbol{c}_1), ..., (\boldsymbol{s}_m, o_m\boldsymbol{c}_m)\}\big). \tag{2}$$

Given the primitive assignment probability vector at image location i, *i.e.*, $\boldsymbol{w}_i = [w_{i1}, w_{i2}, ..., w_{in}]$ predicted by primitive type prediction head, since we consider higher assignment weight leads to the more impact to the rasterized pattern, we can define assignment weight as an approximation of opacity of corresponding transformed primitive, *i.e.*, $w_{ij} \sim o_{ij}$. Hence the final rendered pattern is opacity value weighted combination of all primitives with the same transformation and color attribute (*i.e.*, $(\mathcal{T}_i, \boldsymbol{c}_i)$):

$$\boldsymbol{R}_i = \{(\mathcal{T}_i(\boldsymbol{p}_j), w_{ij}\boldsymbol{c}_i) | j = 1, ..., n\}. \tag{3}$$

Thus, our optimization objective becomes:

$$\min \mathcal{L}\big(\boldsymbol{I}, \boldsymbol{I}^{\mathcal{R}} = \mathcal{R}(\{\boldsymbol{R}_1, ..., \boldsymbol{R}_m\})\big). \tag{4}$$

The above formulation **significantly simplifies the primitive assignment problem** by just softly/continuously adjusting the opacity values during rasterization rather than searching for an optimal assignment matrix in the n^m-size search space. Also, accurate gradients can be propagated to the most likely primitives with the help of differentiable vector graphics rasterization [23].

An example of the comparison between our proposed neural soft assignment and the hard assignment is illustrated in Fig. 3.

3.4. Training Objectives

The image auto-encoder is supervised by an image reconstruction objective, which is written as:

$$\mathcal{L}_r = \|\boldsymbol{I} - \hat{\boldsymbol{I}}\|_2^2, \tag{5}$$

where $\hat{\boldsymbol{I}}$ is the reconstructed pixel image.

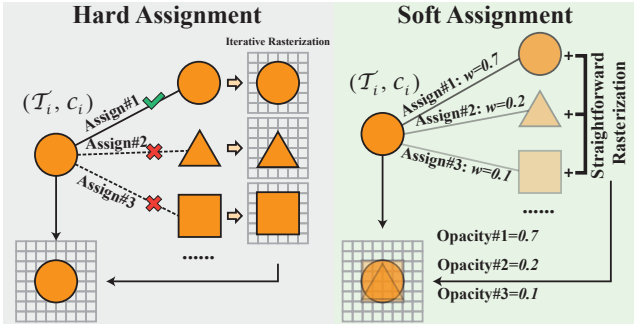

Figure 3. An example of the comparison between our neural soft assignment and the hard assignment.

For the shape attribute set prediction, we rasterize all transformed parametric shapes into a pixel output and compare the result with the input image. More concretely, besides a pixel-wise mean square loss, we follow [41] and utilize a pre-trained image encoder model of CLIP to measure the geometric distance between the input and the rasterized output:

$$\mathcal{L}_a = \|\boldsymbol{I} - \boldsymbol{I}^{\mathcal{R}}\|_2^2 + \mathcal{L}_{geo}, \tag{6}$$

where $\mathcal{L}_{geo}$ can be denoted as:

$$\mathcal{L}_{geo} = \sum_l \|CLIP_l(\boldsymbol{I}) - CLIP_l(\boldsymbol{I}^{\mathcal{R}})\|, \tag{7}$$

where $CLIP_l$ is the CLIP encoder activation at layer l as defined in [41].

Centroid Filtering for Enhancing Gradients. To address the vanishing gradient issue in flat primitive regions, we propose an annealing based region filtering mechanism $\mathcal{F}$ by filtering the image on a Gaussian convolutional kernel G with standard deviation σ: $G(u, v) = \frac{1}{2\pi\sigma^2} e^{-\frac{u^2+v^2}{2\sigma^2}}$. Considering that the rasterized output should gradually approach the original input in later training stage to encourage good reconstruction of detailed structures, we design an annealing strategy to sharp σ as training epoch increases in order to penalize more detailed mismatch: $\sigma_{ep} = \sigma_0 e^{-\frac{ep}{T}}$, where σ_0 is the initial standard deviation of the Gaussian kernel and T controls the annealing rate. Hence the training objective of shape attribute set prediction can be written as:

$$\mathcal{L}_a = \|\mathcal{F}_{ep}(\boldsymbol{I}) - \mathcal{F}_{ep}(\boldsymbol{I}^{\mathcal{R}})\|_2^2 + \mathcal{L}_{geo}, \tag{8}$$

and the overall objective function is:

$$\mathcal{L} = \mathcal{L}_r + \mathcal{L}_a. \tag{9}$$

4. Experiments

4.1. Experimental Setups

Datasets. We evaluate our method on four datasets. Specifically, besides the common Emoji [2] dataset and Icon [1]

dataset introduced in Im2Vec [33], we also introduce a Bauhaus dataset, which contains 200 raster images collected by assembling simple shapes like circles and triangles into Bauhaus-style graphic designs. Compared with Emoji and Icon datasets, the Bauhaus dataset contains more complex and diverse patterns and is more challenging for our task. In addition, to evaluate the generalization of our framework, we augment the Emoji dataset by performing random perspective transformations on the smile faces to obtain more complex image topologies (*i.e.*, Emoji-Aug).

Evaluation Metrics. We evaluate the proposed framework quantitatively on two main tasks including geometric abstraction and image vectorization. For the geometric abstraction task, we use the geometric distance (GD) defined in Eqn. (7) as the metric. We also define an abstraction-based image retrieval task, where the goal is to match the abstraction results with corresponding inputs at instance level. We use image-retrieval accuracy (top-1) as the metric. For the image vectorization task, we use the pixel-wise mean square loss as the metric.

Implementation Details. For transformer network, the feature dimension is 128 and both encoder and decoder have 3 layers. The number of predicted vector graphics paths (*i.e.*, m) for all datasets is set as 10. For the convolutional kernel G in the Centroid Filtering mechanism, the kernel size equals to half of the image size and the standard deviation σ is set as 2.0 initially with an annealing rate $T = 500$. Notably, our framework is scalable and any image reconstruction network that extracts image features can be utilized as our image auto-encoder. All experiments are conducted with a primitive set $\{triangle, rectangle, circle, semicircle\}$, *i.e.*, $n = 4$.

4.2. Comparison with State-of-the-Art Methods

Geometric Abstraction. We compare our framework with state-of-the-arts (*i.e.*, CLIPasso [41] and ES-CLIP [39]) for geometric abstraction task. We report MSE, geometric distance and image retrieval accuracy in Tab. 1. For image retrieval, we use a pre-trained CLIP model to extract features for both original images and corresponding abstraction results. We use a cosine distance function to measure the feature similarity and perform abstraction-based image retrieval at instance level. Considering that the Emoji and Emoji-Aug datasets consist of a large amount of smile faces with similar topologies, instance-level image retrieval is very challenging, especially under top-1 accuracy metric. As seen in Tab. 1, our method achieves significantly better results for all evaluation metrics, especially in geometric distance and retrieval accuracy, which demonstrates that our method predicts precise placement and selection of primitives for geometric abstraction and generates semantically and geometrically discriminative abstraction results.

We also present qualitative comparisons on geometric abstraction in Fig. 4. We see that our method generates compelling abstraction results and fits a wide variety of geometric patterns well. CLIPasso achieves acceptable results by preserving the main semantic concepts but fails to capture fine-grained geometric information, especially in the Icon and Bauhaus datasets where geometric clues are harder to model. The results reveal that sketch-based abstraction cannot explore the diversities of image geometry and it is important to abstract images geometrically with simple shapes. Notably, our method achieves excellent visual results in Bauhaus dataset with only 10 shapes/paths, while ES-CLIP creates great shape redundancy and fails to abstract images effectively with limited number of shapes, which demonstrates that our method realizes efficient and precise geometric abstraction through assembly of predefined primitives and our framework can serve as a useful tool for modern graphic design.

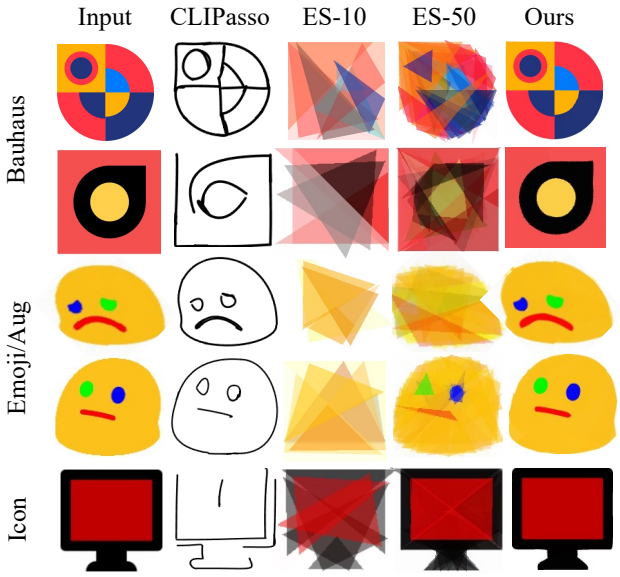

Figure 4. **Qualitative comparison on geometric abstraction task.** All methods except ES-50 use 10 paths to abstract inputs.

Image Vectorization. Considering that our method represents images with vector graphics, we also evaluate the quality of generated vectors and compare with SOTA methods. We report MSE results between inputs and rasterized vector graphics in Tab. 2. Although we only use simple shape primitives rather than complex Bézier curves, we achieve better results on Bauhaus&Icon and comparable results on Emoji compared to state-of-the-art method LIVE [27]. LIVE is a strong method to generate layer-wise vectors for images, but it fails to fit regular geometric shapes well. Also, LIVE is based on single-pass optimization and several minutes are needed to vectorize an image while our method is very efficient. Qualitative comparisons are shown in Fig. 5. We see that our method is capable of generating **regular** vectors to characterize fine-grained image ge-

Method	Bauhaus		Icon			Emoji			Emoji-Aug		
	MSE↓	GD↓	MSE↓	GD↓	Acc.%↑	MSE↓	GD↓	Acc.%↑	MSE↓	GD↓	Acc.%↑
ES-10paths [39]	0.0094	0.514	0.0366	0.382	13.26	0.0306	0.372	7.69	0.0458	0.497	3.90
ES-50paths [39]	0.0046	0.216	0.0235	0.374	40.74	0.0197	0.244	38.64	0.0235	0.312	28.77
CLIPasso [41]	-	0.157	-	0.198	51.85	-	0.228	36.36	-	0.344	30.70
Ours	**0.0003**	**0.015**	**0.0017**	**0.062**	**77.78**	**0.0020**	**0.059**	**76.92**	**0.0022**	**0.072**	**66.23**

Table 1. **Comparison with state-of-the-arts on geometric abstraction task.** We report the mean square error (MSE) in pixel, geometric distance (GD) and image retrieval accuracy (top-1 Acc.%). CLIPasso abstracts images using sketches and ES-CLIP only uses triangles. Both our method and CLIPasso use 10 paths for all datasets while ES-CLIP fails to generate recognizable results under such setting.

Method	Bauhaus	Icon	Emoji	Emoji-Aug
Im2Vec [33]	0.2613	0.3290	0.0258	0.0399
Diffvg [23]	0.0187	0.0285	0.0092	0.0140
LIVE [27]	0.0008	0.0024	**0.0016**	**0.0020**
Ours	**0.0003**	**0.0017**	0.0020	0.0022

Table 2. **Comparison with state-of-the-arts on image vectorization task.** Pixel MSE on several datasets are reported. All results of compared methods are obtained using their source code and all methods use 10 vector graphic paths.

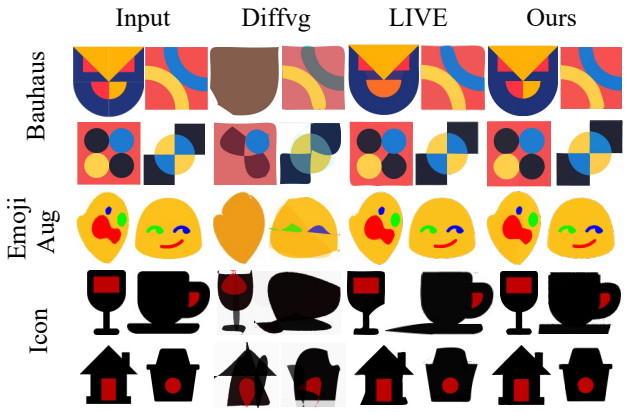

Figure 5. **Qualitative comparison on image vectorization task.** Even though LIVE achieves competitive results in all datasets, it cannot deal with rapid change of color. Also, the vectors generated by LIVE are not as regular as ours, which proves the advantage of primitive-based reconstruction. Please zoom in for more details.

ometry. In addition, existing image vectorization methods are not convenient for image editing while our method has straightforward image editing ability benefited from regular geometric primitives (as shown in Sec. 4.3). Notably, we do not present the results of Im2Vec [33] in Fig. 5 because no acceptable results can be obtained with their source code under 10-paths setting.

4.3. Extension of Our Method

In this section, we explore some valuable extensions of our method. Firstly, motivated by DETR [7] which also utilizes transformer network to perform set prediction, we also visualize the shapes/paths generated by different transformer decoder slots. More concretely, each path prediction is represented as a heatmap centered on its center coordinate and all the heatmaps are added across the dataset according to the path order. The visualization result on Emoji dataset is shown in Fig. 6. We can observe that each decoder slot learns to centralize on specific areas, which motivates two meaningful extensions of our method.

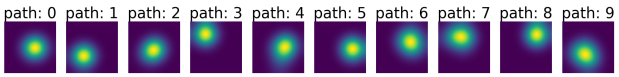

Figure 6. Visualization of the shapes/paths predicted by different transformer decoder slots on the Emoji dataset.

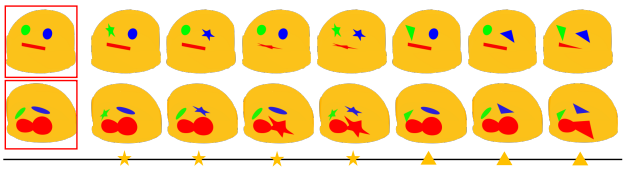

Figure 7. Visualization of geometric style transfer results. We transfer specific semantics of the image to different geometric shapes through simple replacement of primitive type during inference.

Geometric Style Transfer (Editing). Through comparing Fig. 6 and the images of Emoji dataset we can observe that the path 6/7 always generates shape for the right/left eye of the emoji face, and path 9 tends to generate shape for the mouth. Hence, we can perform geometric style transfer (*i.e.*, transfer specific semantics to different geometric shapes) by **directly replacing the corresponding primitive** for specific predicted shapes/paths during inference. For example, star eyes can be achieved by simply replacing the primitives for path 6 and path 7 with a parametric star primitive. Some results of geometric style transfer are shown in Fig. 7. Notably, to edit specific image areas with any parametric shape by simple replacement of primitive during inference, we do not utilize the point-wise offsets stated in Sec. 3.2 and only keep the affine parameters to transform the primitive.

Interpolation. As shown in Fig. 6, the generated shapes of the transformer decoder centralize on specific areas following a uniform order across the dataset, which **overcomes the disordered nature** of previous image vector-

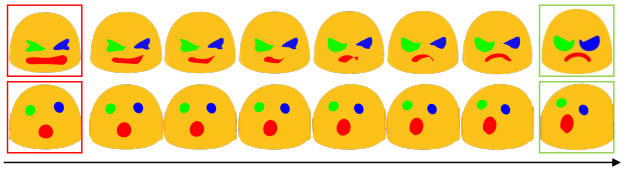

Figure 8. Visualization of interpolation results. It is noted that our method can interpolate not only areas with rich semantics (*e.g.*, eyes and mouth) but also the face poses (bottom row) by simply interpolating the points of the generated shapes in order.

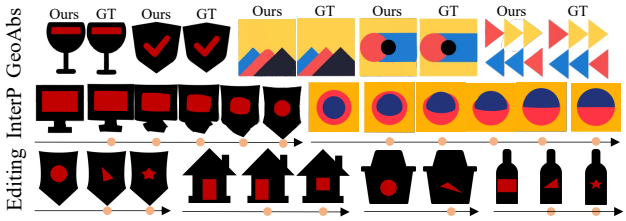

Figure 9. More visualizations on Icon and Bauhaus datasets. 'GeoAbs' means geometric abstraction.'InterP' means vector interpolation. 'Editing' means geometric style transfer by simply modifying primitives during inference.

ization methods. Therefore, our method can perform interpolation between two predicted vector graphics and generate novel art images by simply interpolating the points of each predicted shape in order. Some interpolation results are shown in Fig. 8. We see that our method can interpolate not only areas with rich semantics (*e.g.*, eyes and mouth) but also face poses. Also, we do not utilize the point-wise offsets and only keep the affine parameters in this part.

4.4. Ablation Study

In this section, we explore the crucial components and hyper-parameters of our method. For simplicity, all the experiments in this section are conducted on Emoji dataset.

Component Analyses. We first investigate the effectiveness of the proposed neural soft assignment scheme. We compare our soft assignment algorithm with several ablated versions and the corresponding MSE losses between input and the rasterized output during training process are shown in Fig. 10(a). We see that without establishing equivalence between primitive selection and the opacity for rasterization (*i.e.*, Baseline), the final MSE loss converges at a high value and it is not able to generate accurate rasterized results because of the lack of supervision for the primitive selection. The assignment weights can also be optimized by using *argmax* to select primitives (*i.e.*, HardAssign). However, the hard assignment transfer process is difficult to optimize because of unavailable pairwise assignment cost and the non-differentiable nature of the *argmax* function. Gumbel Softmax can be utilized to provide more accurate gradients when using *argmax* function, but it attaches great

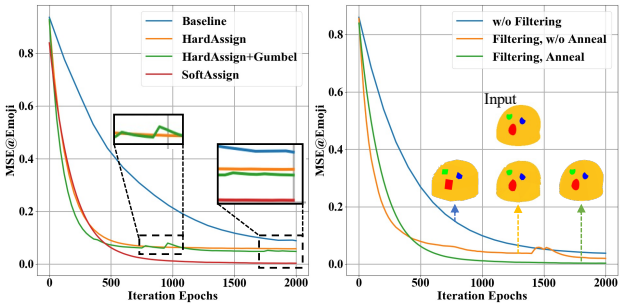

Figure 10. **Component Analyses.** (a) The effectiveness of our neural soft assignment scheme is illustrated. (b) The effectiveness of both image filtering and annealing strategy is demonstrated. Please zoom in for more detailed comparisons.

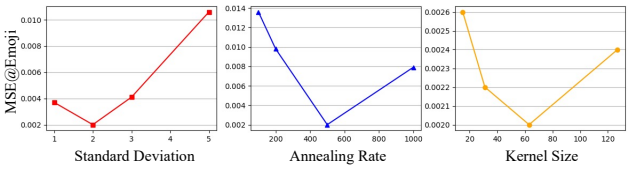

Figure 11. **Parameter Analyses.** All experiments are conducted on Emoji dataset with MSE metric.

randomness to primitive selection and makes the optimization for transformation parameters unstable (*i.e.*, HardAssign+Gumbel). Our neural soft assignment scheme can perform stable and accurate optimization by reducing the optimization difficulty and providing accurate gradients for assignment weights. We also explore the effectiveness of our centroid filtering mechanism. The results are shown in Fig. 10(b). We observe that the image loss converges faster and better by enhancing the gradients using annealing mechanism. Without the centroid filtering mechanism, it is able to transform primitives to appropriate placements with corresponding colors but fails to further adjust the shape to fit the image geometry due to the vanishing gradient in flat primitive regions. Also, there exists visible pixel-level blur when not using annealing strategy.

Parameter Analyses. We explore how initial standard deviation σ_0, the kernel size of the Gaussian kernel and the annealing rate T in the centroid filtering mechanism affect our model. The results are shown in Fig. 11. We observe that too large σ_0 and too small T cause poor results and our method is not sensitive to the kernel size (all MSEs are below 0.003). We also explore the effect of the number of shapes/paths m and the point-wise offsets on the results. The results are shown in Fig. 12. Our method can generate competitive results with only 5 shapes on Emoji dataset. We also observe that the point-wise offsets can reduce artifacts among shapes and are very effective especially when the number of shapes is limited.

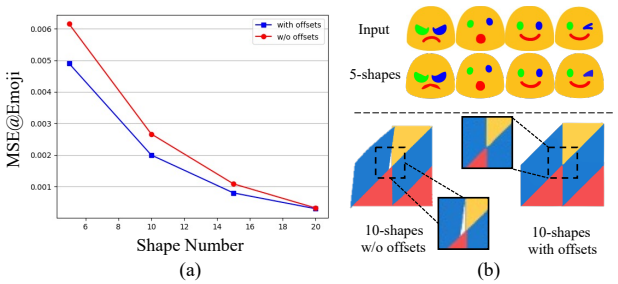

Figure 12. MSE results *vs.* shape number on Emoji dataset are shown. Competitive results are generated with only 5 shapes. It is also illustrated that our method can achieve better results equipped with point-wise offsets. Please zoom in for more details.

5. Conclusion

In this work, we present a novel image geometric abstraction paradigm based on assembly out of a pool of predefined parametric primitives in an unsupervised manner. Experimental results on various datasets and tasks demonstrate that our framework generates accurate and high-quality/regular vectors to characterize geometric information of images with combination of only four simple primitives, facilitating straightforward controllable shape editing by simple replacement of primitive type during inference.

6. Acknowledgment

This work was supported by National Science Foundation of China (U20B2072, 61976137). This work was also partially supported by Grant YG2021ZD18 from Shanghai Jiaotong University Medical Engineering Cross Research.

References

[1] creativestall. `https://thenounproject.com/creativestall/`.

[2] Note emoji. `https://github.com/googlefonts/noto-emoji`. Accessed: 2021-09-30.

[3] Stephan Alaniz, Massimiliano Mancini, Anjan Dutta, Diego Marcos, and Zeynep Akata. Abstracting sketches through simple primitives. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIX*, pages 396–412. Springer, 2022.

[4] Mohammad Ali Alomrani, Reza Moravej, and Elias B Khalil. Deep policies for online bipartite matching: a reinforcement learning approach. *arXiv preprint arXiv:2109.10380*, 2021.

[5] Hans-Georg Beyer and Hans-Paul Schwefel. Evolution strategies–a comprehensive introduction. *Natural computing*, 1:3–52, 2002.

[6] Irving Biederman and Ginny Ju. Surface versus edge-based determinants of visual recognition. *Cognitive psychology*, 20(1):38–64, 1988.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[9] Magdalena. Dabrowski. Geometric abstraction. In Heilbrunn Timeline of Art History. New York: The Metropolitan Museum of Art, 2000. `http://www.metmuseum.org/toah/hd/geab/hd_geab.htm`.

[10] James Richard Diebel. *Bayesian Image Vectorization: the probabilistic inversion of vector image rasterization*. Stanford University, 2008.

[11] Kevin Frans, Lisa B Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*, 2021.

[12] Yaroslav Ganin, Sergey Bartunov, Yujia Li, Ethan Keller, and Stefano Saliceti. Computer-aided design as language. *Advances in Neural Information Processing Systems*, 34:5885–5897, 2021.

[13] James J Gibson. A theory of pictorial perception. *Audiovisual communication review*, 2(1):3–23, 1954.

[14] James J Gibson. The information available in pictures. *Leonardo*, 4(1):27–35, 1971.

[15] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017.

[16] Paul Haeberli and Mark Segal. Texture mapping as a fundamental drawing primitive. In *Fourth Eurographics Workshop on Rendering*, volume 259, page 266, 1993.

[17] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[18] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02):107–116, 1998.

[19] Xuemei Hu, Jinli Suo, Tao Yue, Liheng Bian, and Qionghai Dai. Patch-primitive driven compressive ghost imaging. *Optics express*, 23(9):11092–11104, 2015.

[20] Moritz Kampelmuhler and Axel Pinz. Synthesizing human-like sketches from natural images using a conditional convolutional decoder. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3203–3211, 2020.

[21] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[22] 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.

[23] Tzu-Mao Li, Michal Lukáč, Michaël Gharbi, and Jonathan Ragan-Kelley. Differentiable vector graphics rasterization for editing and learning. *ACM Transactions on Graphics (TOG)*, 39(6):1–15, 2020.

[24] Yi Li, Yi-Zhe Song, Timothy M Hospedales, and Shaogang Gong. Free-hand sketch synthesis with deformable stroke models. *International Journal of Computer Vision*, 122:169–190, 2017.

[25] Jinxian Liu, Ye Chen, Bingbing Ni, Jiyao Mao, and Zhenbo Yu. Inferring fluid dynamics via inverse rendering. *arXiv preprint arXiv:2304.04446*, 2023.

[26] Jinxian Liu, Bingbing Ni, Ye Chen, Zhenbo Yu, and Hang Wang. Learning by restoring broken 3d geometry. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[27] Xu Ma, Yuqian Zhou, Xingqian Xu, Bin Sun, Valerii Filev, Nikita Orlov, Yun Fu, and Humphrey Shi. Towards layer-wise image vectorization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16314–16323, 2022.

[28] Umar Riaz Muhammad, Yongxin Yang, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Learning deep sketch abstraction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8014–8023, 2018.

[29] Wamiq Para, Shariq Bhat, Paul Guerrero, Tom Kelly, Niloy Mitra, Leonidas J Guibas, and Peter Wonka. Sketchgen: Generating constrained cad sketches. *Advances in Neural Information Processing Systems*, 34:5077–5088, 2021.

[30] Todd B Pittman, YH Shih, DV Strekalov, and Alexander V Sergienko. Optical imaging by means of two-photon quantum entanglement. *Physical Review A*, 52(5):R3429, 1995.

[31] Yonggang Qi, Guoyao Su, Pinaki Nath Chowdhury, Mingkang Li, and Yi-Zhe Song. Sketchlattice: Latticed representation for sketch manipulation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 953–961, 2021.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[33] Pradyumna Reddy, Michael Gharbi, Michal Lukac, and Niloy J Mitra. Im2vec: Synthesizing vector graphics without vector supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7342–7351, 2021.

[34] 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.

[35] Ari Seff, Wenda Zhou, Nick Richardson, and Ryan P Adams. Vitruvion: A generative model of parametric cad sketches. *arXiv preprint arXiv:2109.14124*, 2021.

[36] 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.

[37] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.

[38] David S Taubman, Michael W Marcellin, and Majid Rabbani. Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging*, 11(2):286–287, 2002.

[39] Yingtao Tian and David Ha. Modern evolution strategies for creativity: Fitting concrete images and abstract concepts. In *Artificial Intelligence in Music, Sound, Art and Design: 11th International Conference, EvoMUSART 2022, Held as Part of EvoStar 2022, Madrid, Spain, April 20–22, 2022, Proceedings*, pages 275–291. Springer, 2022.

[40] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[41] 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 Transactions on Graphics (TOG)*, 41(4):1–11, 2022.

[42] Yansheng Wang, Yongxin Tong, Cheng Long, Pan Xu, Ke Xu, and Weifeng Lv. Adaptive dynamic bipartite graph matching: A reinforcement learning approach. In *2019 IEEE 35th international conference on data engineering (ICDE)*, pages 1478–1489. IEEE, 2019.

[43] Jie Wu, Changhu Wang, Liqing Zhang, and Yong Rui. Offline sketch parsing via shapeness estimation. In *IJCAI*, volume 15, pages 1200–1206. Citeseer, 2015.

[44] Tian Xia, Binbin Liao, and Yizhou Yu. Patch-based image vectorization with automatic curvilinear feature alignment. *ACM Transactions on Graphics (TOG)*, 28(5):1–10, 2009.

[45] Changcheng Xiao, Changhu Wang, Liqing Zhang, and Lei Zhang. Sketch-based image retrieval via shape words. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*, pages 571–574, 2015.

[46] Guofu Xie, Xin Sun, Xin Tong, and Derek Nowrouzezahrai. Hierarchical diffusion curves for accurate automatic image vectorization. *ACM Transactions on Graphics (TOG)*, 33(6):1–11, 2014.

[47] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 574–591. Springer, 2020.

[48] Fenggen Yu, Zhiqin Chen, Manyi Li, Aditya Sanghi, Hooman Shayani, Ali Mahdavi-Amiri, and Hao Zhang. Capri-net: learning compact cad shapes with adaptive primitive assembly. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11768–11778, 2022.