Sketch-BERT: Learning Sketch Bidirectional Encoder Representation from Transformers by Self-supervised Learning of Sketch Gestalt

Hangyu Lin*, Yanwei Fu *
School of Data Science, Fudan University
18210980008,yanweifu@fudan.edu.cn

Yu-Gang Jiang†, Xiangyang Xue
School of Computer Science, Fudan University
ygj, xyxue@fudan.edu.cn

Abstract

Previous researches of sketches often considered sketches in pixel format and leveraged CNN based models in the sketch understanding. Fundamentally, a sketch is stored as a sequence of data points, a vector format representation, rather than the photo-realistic image of pixels. SketchRNN [7] studied a generative neural representation for sketches of vector format by Long Short Term Memory networks (LSTM). Unfortunately, the representation learned by SketchRNN is primarily for the generation tasks, rather than the other tasks of recognition and retrieval of sketches. To this end and inspired by the recent BERT model [3], we present a model of learning Sketch Bidirectional Encoder Representation from Transformer (Sketch-BERT). We generalize BERT to sketch domain, with the novel proposed components and pre-training algorithms, including the newly designed sketch embedding networks, and the self-supervised learning of sketch gestalt. Particularly, towards the pre-training task, we present a novel Sketch Gestalt Model (SGM) to help train the Sketch-BERT. Experimentally, we show that the learned representation of Sketch-BERT can help and improve the performance of the downstream tasks of sketch recognition, sketch retrieval, and sketch gestalt.

1. Introduction

With the prevailing of touch-screen devices, *e.g.*, iPad, everyone can easily draw simple sketches. It thus supports the demand of automatically understanding the sketches, which have been extensively studied in [28, 22, 17] as a type of 2D pixel images. Interestingly, the free-hand sketches reflect our abstraction and iconic representation that are composed of patterns, structure, form and even simple logic of objects and scenes in the world around us. Thus rather than being taken as 2D images, sketches should be intrinsically analyzed from the view of sequential data, which however, has less been touched in earlier works. Typically, a sketch consists of several strokes where each stroke can be seen as a sequence of points. We take the same 5-element vector format representation for sketches as in [7]. Briefly speaking, each point has 2-dimensional continuous position value and 3-dimensional one hot state value which indicates the state of the point.

According to Gestalt principles of perceptual grouping [2], humans can easily perceive a sketch as a sequence of data points. To analyze the sequential sketch drawings, SketchRNN [7] aimed at learning neural representation of sketches by combining variational autoencoder (VAE) with a Long Short Term Memory networks (LSTM), primary for the sketch generation. In contrast, human vision systems would be capable of both understanding semantics, or abstracting the patterns from sketches. For instance, we can easily both predict the category label of sketches from "Ground Truth" column (sketch recognition task), and complete the "Masked Input" column of sketches (sketch gestalt task), as shown in Fig. 1. Comparably, this demands significant high quality in learning much more general and comprehensive sketch representation.

Formally, a new sketch Gestalt (sGesta) task is, for the first time, proposed in this paper as in Fig. 1. The name *sketch Gestalt* comes from the famous Gestalt theory which emphasizes the whole structure of an object rather than some parts. Particularly, the task of sketch gestalt aims at recovering the masked parts of points in sketches and completes the shape of masked sketches. It needs to predict both continuous position values and discrete state values which are utilized to define the sketch points. We show that leveraging the sketch gestalt task helps better understanding the general patterns of sketches.

To this end, this paper proposes a novel model of learning Sketch Bidirectional Encoder Representation from Transformer (Sketch-BERT), which is inspired by the recent BERT model [3] from Natural Language Processing (NLP). Essentially, the transformer structure exerts great

*indicates equal contributions, † indicates corresponding author. Y. Fu is with School of Data Science, and MOE Frontiers Center for Brain Science, Shanghai Key Lab of Intelligent Information Processing Fudan University.

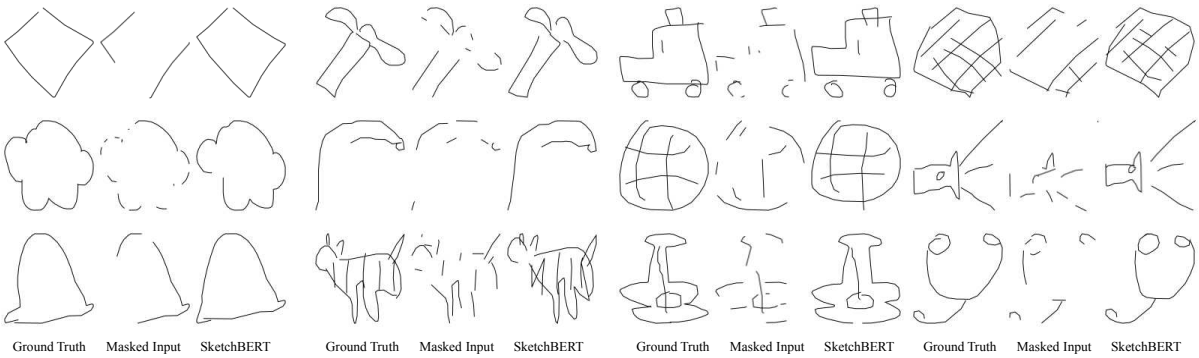


Figure 1. Sketch Gestalt which aims at recovering the masked parts of points in sketches and complete the shape of masked sketches.

potential in modeling the sequential data; and we adopt the weight-sharing multi-layer transformer structure from [16], which share the merits of BERT and yet with much less total parameters. Particularly, a novel embedding method is tailored for sketches, and encodes three level embeddings, *i.e*., point, positional, and stroke embedding. A refinement embedding network is utilized to project the embedding features into the input feature space of transformer.

To efficiently train our Sketch-BERT, we introduce a novel task – self-supervised learning by sketch gestalt, which includes the targets of mask position prediction, and mask state prediction. Correspondingly, we further present in addressing these tasks, a novel Sketch Gestalt Model (SGM), which is inspired by the Mask Language Model in NLP. The pre-trained Sketch-BERT is capable of efficiently solving the learning tasks of sketches. Particularly, this paper considers the tasks of sketch recognition, sketch retrieval, and sketch gestalt.

Contributions. We make several contributions in this paper. (1) The BERT model is extended to sketches, that is, we for the first time, propose a Sketch-BERT model in efficiently learning neural representation of sketches. Critically, our Sketch-BERT has several novel components, which are significant different from the BERT model, including the novel three-level embedding for sketches, and self-supervised learning by sketch gestalt. (2) To the best of our knowledge, a novel task – sketch Gestalt (sGesta) is for the first time studied in this paper. This task is inspired by the Gestalt principles of perceptual grouping. (3) A self-supervised learning process by sketch gestalt, is presented. Empirically, we show that the corresponding SGM for this task can efficiently help pre-train our Sketch-BERT, and thus significantly boost the performance of several downstream sketch tasks.

2. Related Works

Representation of Sketches. The research on representation of sketches has been lasted for a long time. As the studies of images and texts, learning discriminative feature for sketches is also a hot topic for learning sketch representation. The majority of such works [11, 19, 28, 27, 20, 17] achieved the goal through the classification or retrieval tasks. Traditional methods always focused on hand-crafted features, such as BoW [11], HOG [10] and ensemble structured features [19]. Recently, there are works that tried to learn neural representation of sketches. Due to the huge visual gap between sketches and images, Sketch-A-Net [28] designed a specific Convolutional Neural Network (CNN) structure for sketches, which achieved the state-of-art performance at that time, with several following works [27, 22]. On the other hand, TC-Net [20] utilized an auxiliary classification task to directly solve the sketch recognition by the backbone, *e.g.*, DenseNet [12]. Different from the above methods which directly utilized the pixel level information from sketch images, researchers made use of vector form representation of sketches in [17, 30].

Generation and Gestalt of Sketch. Sketch generation, as another significant topic for learning sketches, also draws more and more attention. In [14, 32, 18], they generated sketches from images via convolutional neural networks and translation losses. SketchRNN [7] employed LSTM to solve both conditional and unconditional generation on vector images of sketches. Reinforcement learning-based models [31, 13] also worked well on learning stroke-wise representation from pixel images of sketches. Besides the generation task, we propose a new sketch gestalt task in this paper. Despite this task shares the same goal as image inpainting in completing the masked regions/parts, the key differences come from several points, including, (1) the models for image inpainting [26, 25] mostly predict pixels by existing parts in images; in contrast, sketch gestalt aims at recovering the abstract shapes of some objects. (2) the texture, color and background information are utilized to help image inpainting models maintain the visual consistency of whole images, while more abstract information, e.g., shape, would be more advisable for sketches in completing the abstraction and iconic sketches.

Transformers and Self-supervised Learning. Beside CNN models, it is essential to learn sequence models for learning how to represent sketches. Recurrent neural networks [9, 1] are the most successful sequential models during the last decades. Recently, researchers believe that "attention is all your need" [23]; and the models based on Transformer are dominating the performance on almost all NLP tasks. Particularly, BERT [3] exploited the mask language model as pre-training task. Further XLNet [24] generalized the language modeling strategy in BERT. Such models are all trained in a self-supervised way and then fine-tuned on several downstream tasks. Inspired by this, we design a novel self-supervised learning method for sketches which can help Sketch-BERT understand the structure of sketches.The task of self-supervised learning [15] is generally defined as learning to predict the withheld parts of data. It thus forces the network to learn what we really care about, such as, image rotation [6], image colorization [29], and jigsaw puzzle [21]. However, most of previous self-supervised learning models are specially designed for images, rather than the sketch. Comparably, the first self-supervised learning by sketch gestalt is proposed and studied in this paper.

.

3. Methodology

This section introduces our Sketch-BERT model and the learning procedure. Particularly, our model embeds the input sketch as a sequence of points. A weight-sharing multi-layer transformer is introduced for sketches, and thus it performs as the backbone to our Sketch-BERT. A novel self-supervised learning task – sketch Gestalt task, is proposed to facilitate training Sketch-BERT.

3.1. Embedding Sketches

Generally, a sketch is stored as a sequential set of strokes, which is further represented as a sequence of points. As the vector data format in [7], a sketch can be represented as a list of points, where each point contains 5 attributes,

$$(\Delta x, \Delta y, p_1, p_2, p_3) \tag{1}$$

where Δx and Δy are the values of relative offsets between current point and previous point; (p_1, p_2, p_3) would be utilized as a one-hot vector indicating the state of each point ($\sum_{i=1}^{3} p_i = 1$); $p_2 = 1$ indicates the ending of one stroke; $p_3 = 1$ means the ending of the whole sketch, and $p_1 = 1$ represents the other sequential points of sketches. We normalize the position offsets of each point by dividing the maximum offset values, and make sure $\Delta x, \Delta y \in [0, 1]$.

Point Embedding. Sketches are then embedded as the sequential representation to learn Sketch-BERT. The point information $(\Delta x, \Delta y, p_1, p_2, p_3)$ is learned as an embedding

$$\mathrm{E}_{pt} = W_{pt} \left(\Delta x, \Delta y, p_1, p_2, p_3\right)^T \tag{2}$$

where $W_{pt} \in R^{d_E \times 5}$ is the embedding matrix, and d_E is the dimension of the point embedding.

Positional Embedding. The position of each sequential point should be encoded; and thus we introduce the positional embedding with learnable embedding weight W_{ps},

$$\mathrm{E}_{ps} = W_{ps}\mathbf{1}_{ps} \in R^{d_E} \tag{3}$$

where $\mathbf{1}_{ps}$ is one-hot positional vector. In particular, we set the max length of each sketch sequence up to 250, while remove the points of the sequence beyond 250, by default.

Stroke Embedding. We also learn to embed the sequences of strokes. Inspired by the segment embedding in language model [3], the strokes of sketch are also embedded as

$$\mathrm{E}_{str} = W_{str}\mathbf{1}_{str} \in R^{d_E} \tag{4}$$

with the length of stroke sequence up to 50; where $\mathbf{1}_{str}$ is corresponding one-shot stroke vector. Thus, we have the following final sketch embedding as,

$$\mathrm{E} = \mathrm{E}_{pt} + \mathrm{E}_{ps} + \mathrm{E}_{str} \tag{5}$$

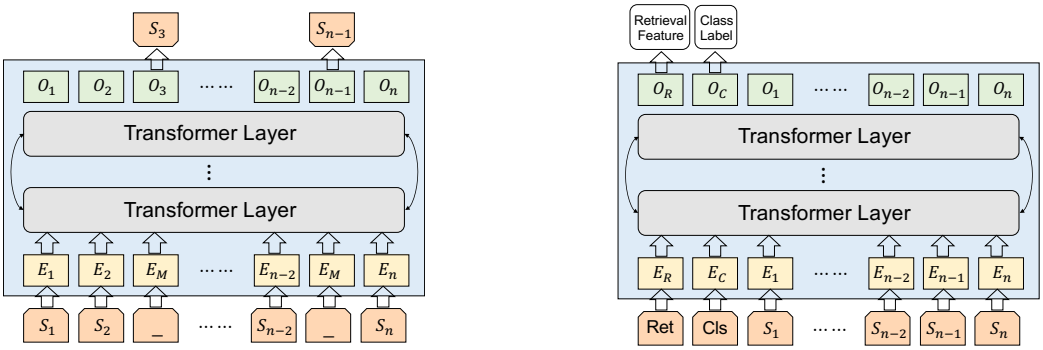
Refine Embedding Network. We further employ a refine embedding network to improve the embedding dimension from d_E to d_H, used in the transformer. Specifically, the refine embedding network consists of several fully-connected layers with the input and output dimensions d_E and d_H, respectively. In our Sketch-Bert, we have $d_E = 128, d_H = 768$, and the structure of refinement network is $128 - 256 - 512 - 768$, where the neurons of two hidden layers are 256 and 512, respectively.

3.2. Weight-sharing Multi-layer Transformer

We adopt the weight-sharing multi-layer bidirectional transformer as the backbone, inspired by the ALBERT [16] and BERT [3]. Particularly, the weights are shared in the layers of the encoder. This makes a faster convergence of Sketch-BERT. Formally, we denote the sketch embedding as

$$\mathrm{E} = (\mathrm{E}_1, \mathrm{E}_2, \cdots, \mathrm{E}_n) \in R^{n \times d_H}$$

where n is the true length of each sketch embedding. Hidden features will be updated by self-attention module in each weight-sharing transformer layer. The final output features from the Sketch-BERT encoder will be used for different downstream tasks.



(a) Sketch-BERT for Sketch Gestalt

(b) Sketch-BERT for Sketch Recognition/Retrieval

Figure 2. Overview structure of Sketch-BERT for Sketch Gestalt Model and downstream tasks.

3.3. Self-Supervised Learning by Sketch Gestalt

Since the pre-training tasks over unlabeled text data in NLP have shown great potential in improving the performance of BERT, it is essential to introduce a self-supervised learning task to facilitate pre-training our Sketch-BERT.

To this end, we define a novel self-supervised learning process by sketch Gestalt (sGesta), which aims at recovering the masked points in sketches as shown in Fig. 2(a). Given a masked sketch in vector format $s_{mask} = s_{gt} \cdot m$ where m is the mask with the same shape of s_{gt}, sketch Gestalt targets at predicting s_{comp} which has the same shape and semantic information as s_{gt} from the s_{mask}. Specifically, the position mask at first two dimensions and state mask at other dimensions can be predicted, individually. To solve the self-supervised learning task, we present the *Sketch Gestalt Model* (SGM). As in Eq (1), each point is represented by the key information of both positional offset $(\Delta x, \Delta y)$ and state (p_1, p_2, p_3), which will be masked and predicted by our SGM, individually. We propose different mask strategies for positional offset and state information to help train our Sketch-BERT. By default, we mask 15% of all positions and states respectively for each sketch sequence.

Mask Position Prediction. We divide the offset values for points into two classes: 1) the offset for a point in a stroke; 2) the offset for a point as the start of a stroke. In sketches, distributions of these two type offset values are quite different, and there are also total distinctive value ranges of two types of offset values. Thus we generate the masks by sampling points in these two classes, proportional to the total point number of each point type class, by setting $(\Delta x, \Delta y)$ of the masked point to 0.

Mask State Prediction. Quite similarly, there are imbalance distributions of p_1, p_2, p_3 for sketch points. In particular, there are always much more points with p_1 than those with p_2 or p_3. Thus, we mask the state of each point, in term of the percentage of points with the state p_1, p_2, p_3. If the state of one point is masked, it has $p_1 = p_2 = p_3 = 0$.

Embedding Reconstruction Network. Our SGM introduces an embedding reconstruction network, which plays the corresponding decoder of the refine embedding network. In particular, given as the input the d_H dimensional embedding features, the reconstruction network predicts the states and positions of each mask. Practically, we reverse the structure of refine embedding network, and utilize the structure as $768 - 512 - 256 - 128 - 5$, with the neurons of 512, 256, and 128 of hidden layers, individually. We adopt L_1 loss for mask position prediction, to predict the continuous position offset values; and, we use the standard cross entropy loss for different state categories in mask state prediction.

3.4. Learning Tasks by Sketch-BERT

We further elaborate how Sketch-BERT model could be utilized for different downstream tasks after the pre-training procedure by the self-supervised learning. For each task, we give the formal definition and describe how the pre-trained Sketch-BERT model can be utilized here. Especially, we are interested in following tasks.

Sketch Recognition. This task takes a sketch s as input and predicts its category label c. To fine-tune the Sketch-BERT for recognition task, we add a [CLS] label, *i.e.*, a special token to the beginning of the sequential data of each sketch, as shown in Fig. 2(b). For recognition tasks, our Sketch-BERT serves as a generic feature extractor of each sketch. A standard softmax classification layer as well as cross entropy loss, is applied to the outputs of Sketch-BERT (O_C). The training sketches of recognition tasks have been utilized to fine-tune the Sketch-BERT, and train the classification layer, as the standard practice in BERT [3].

Sketch Retrieval. Given a query sketch s_q, sketch retrieval task targets at finding sketches $s_1, \ldots, s_n$ with the same category as the query s_q. We add the [RET] label token to

the beginning of sequential data of each sketch, and use the Sketch-BERT to extract the features (O_R) of each sketch, as in Fig. 2(b). To conduct the retrieval task, the output features are projected into a fully connected layer of 256 neurons, which is optimized by a triplet loss as in [20] by minimizing the distance of sketches in the same class, and maximizing the distance of sketches in different classes. In addition, we also apply the cross entropy loss of learning to predict the category of each sketch. The training data of retrieval task is utilized to train the newly added fully connected layer, and fine-tune the Sketch-BERT.

Sketch Gestalt. Inspired by the Gestalt principles of perceptual grouping, this task is introduced to recover a realistic sketch images s_{comp} given an incomplete s_{mask} as shown in Fig. 2(a). We directly utilize the SGM learned in self-supervised learning step for this task.

4. Experiments and Discussion

4.1. Datasets and Settings

Datasets. Our model is evaluated on two large-scale sketch datasets – QuickDraw dataset [7], and TU-Berlin dataset [4] (1) QuickDraw dataset is collected from Google application *Quick, Draw!*, an online game to draw a sketch less than 20 seconds. There are about 50 million sketch drawings across total 345 classes of common objects. Here we follow the pre-process method and training split from [7], where each class has 70K training samples, 2.5K validation and 2.5K test samples in QuickDraw dataset. We also simplify the sketches by applying the Ramer-Douglas-Peucker (RDP) algorithm, leading to a maximum sequence length of 321. (2) TU-Berlin contains less quantity but better quality sketch samples than QuickDraw. There are 250 object categories in TU-Berlin with 80 sketches in each category.

Implementation Details. In our work, the Sketch-BERT model has $L = 8$ weight-sharing Transformer layers with the hidden size of $H = 768$ and the number of self-attention heads of 12. The same with BERT, the feed-forward size will be set to $4H$ in the weight-sharing transformer layer. The embedding size is set to 128 and the refine embedding network is a fully-connected network of neurons $128 - 256 - 512 - 768$. Correspondingly, the reconstruction network is composed of four fully-connected layers of neurons $768 - 512 - 256 - 128 - 5$. The max lengths of input sketches are set as 250, and 500 for QuickDraw, and TU-Berlin, respectively. We implement our Sketch-BERT model with PyTorch. To optimize the whole model, we adopt Adam optimizer with a learning rate of 0.0001. In self-supervised learning, we leverage the whole training data from QuickDraw to train the sketch gestalt model.

Competitors. We compare several baselines here. (1) HOG-SVM [5]: It is a traditional method utilized HOG feature and SVM to predict the classification result. (2) Ensemble [19]: This model leverages several types of features for sketches, we evaluate it on classification task. (3) Bi-LSTM [9] : We employ a three-layer bidirectional LSTM model to test the recognition and retrieval tasks on sequential data of sketches. The dimension of the hidden states is set to 512 here. (4) Sketch-a-Net: [28]: The Sketch-a-Net is a specifically designed convolutional neural network for sketches. (5) DSSA[22] add an attention module and a high-order energy triplet loss function to original Sketch-A-Net model. (6) ResNet: We also evaluate residual network, one of the most popular convolutional neural network in computer vision field designed for image recognition task. (7) TC-Net [20]: It is a network based on DenseNet [12] for sketch based image retrieval task, we leverage the pre-trained model for classification and retrieval tasks. (8) SketchRNN [7]: SketchRNN employed a variational autoencoder with LSTM network as encoder and decoder backbones to solve the sketch generation task, in our experiments, we use this approach to test the sketch gestalt task. The training and validation set of datasets are employed to train our models and competitors, which are further validated in the test set. For fair comparison of structure, we retrain all models on QuickDraw and TU-Berlin datasets for different tasks.

Methods	QuickDraw (%)		TU-Berlin (%)	
	T-1	T-5	T-1	T-5
HOG-SVM [4]	56.13	78.34	56.0	–
Ensemble [19]	66.98	89.32	61.5	–
Bi-LSTM [9]	86.14	97.03	62.35	85.25
Sketch-a-Net* [27]	–	–	**77.95**	–
Sketch-a-Net [27]	75.33	90.21	47.70	67.00
DSSA [22]	79.47	92.41	49.95	68.00
ResNet18 [8]	83.97	95.98	65.15	83.30
ResNet50 [8]	86.03	97.06	69.35	90.75
TCNet [20]	86.79	97.08	73.95	91.30
Sketch-BERT (w./o.)	83.10	95.84	54.20	66.05
Sketch-BERT (w.)	**88.30**	**97.82**	76.30	**91.40**

Table 1. The Top-1 (T-1) and Top-5 (T-5) accuracy of our model and other baselines on classification task; w./o., and w. indicate the results without, and with the self-supervised learning by sketch Gestalt, individually. * means the results in original paper [27].

4.2. Results on Sketch Recognition Task

Recognition or classification is a typical task for understanding or modeling data in term of semantic information, so we first compare the classification results of our model with other baselines. We use 100 categories with 5K train samples, 2.5K validation samples and 2.5K test samples for QuickDraw dataset; whole categories of TU-Berlin dataset with training split of $80\%/10\%/10\%$ for train/validation/test samples, respectively.

Models	QuickDraw			TU-Berlin		
	Top-1 (%)	Top-5 (%)	mAP (%)	Top-1 (%)	Top-5 (%)	mAP(%)
Bi-LSTM [9]	70.91	89.52	60.11	31.40	59.60	23.71
Sketch-a-Net [27]	74.88	90.10	65.13	37.25	63.50	26.18
DSSA [22]	78.16	91.04	68.10	38.45	66.10	28.77
ResNet18 [8]	80.34	91.71	70.98	41.45	67.10	29.33
ResNet50 [8]	82.41	92.52	74.84	51.80	74.45	36.94
TCNet [20]	83.59	92.57	76.38	55.30	79.45	38.78
Sketch-BERT (w./o.)	63.13	84.70	55.10	32.50	57.90	24.14
Sketch-BERT (w.)	**85.47**	**93.49**	**78.87**	**57.25**	**81.50**	**41.54**

Table 2. The Top-1, Top-5 accuracy and mean Average Precision(mAP) of our model and other baselines on sketch retrieval task. w./o., and w. indicate the results without, and with the self-supervised learning by sketch gestalt.

From the results in Tab. 1, it is obvious that the Sketch-BERT outperforms other baselines including both pixel images based models like Sketch-a-Net, ResNet18/50 or TC-Net; and vector images based model like Bi-LSTM by a considerable margin: about 2% on QuickDraw. This indicates the effectiveness of our Sketch-BERT model, and self-supervised pipeline by sGesta. Particularly, we give the ablation study of our Sketch-BERT without using self-supervised training (i.e., Sketch-BERT (w./o.) in Tab. 1). It gives us the results of 5% dropping of top-1 accuracy on QuickDraw dataset. In fact, this can reveal the power of our SGM proposed in this paper. Furthermore, the Sketch-BERT (w.) gets converged much faster than that of Sketch-BERT (w./o.) if they are fine-tuned on the same training data. For example, the convergence epoch reduces from 50 epochs of Sketch-BERT (w./o.), to only 5 epochs of Sketch-BERT (w.), for the recognition task trained on TU-Berlin dataset.

4.3. Results on Sketch Retrieval Task

We are particularly interested in the category-level sketch retrieval and test sketch retrieval task over the same dataset as the recognition task. To evaluate the performance of different models, we report both Top-1/5 accuracy and mean Average Precision (mAP). To make a fair comparison to the other baselines We employ the typical triplet loss and cross entropy loss, as our Sec. 3.4. Each model only serves as the backbone to extract the sketch features from the a tuple of anchor sketch, positive sketch, negative sketch. The ranked retrieval results are compared.

The results are summarized in Tab. 2. Our Sketch-BERT model with self-supervised learning tasks has a much higher performance than the other baselines. It gives us about 2% improvement over the best second method — TCNet, which is the state-of-the-art CNN based model for sketch recognition. We notice that the vector based model – Bi-LSTM only achieves 70% top-1 accuracy, while the others CNN based models get the performance over 75% accuracy. On the other hand, interestingly our Sketch-BERT without self-supervised training by sGesta, achieves much worse results than the other baselines on this retrieval task. This further suggests that our SGM model proposed in self-supervised learning step, can efficiently improve the generalization ability of our Sketch-BERT. To sum up, the results from both sketch classification and sketch retrieval tasks show the superiority of our Sketch-BERT model on the sketch representation learning.

4.4. Results on Sketch Gestalt Task

Rather than discriminative neural representation, Sketch-BERT model also has a good capacity for generative representation like sketch gestalt task, where some part of sketches have been masked, and predicted by the models. In this section, our model is compared against SketchRNN [7], which, to the best of our knowledge, is the only generative model that is able to predict the masked sketch sequences. This task is conducted on QuickDraw dataset: both models are learned on training data, and predicted on the test data.

We illustrate some completed results from several classes in QuickDraw dataset in Fig. 3. The four columns in the figure represent (1) ground truth sketch, (2) incomplete or masked input with a random 30% mask on position and state together, (3) completed results from the SketchRNN, (4) completed results from our Sketch-BERT model.

We can show that our Sketch-BERT model has a much better ability in understanding and filling the masked sketches in a more correct way than that of SketchRNN. Particularly, we further analyze and compare these results. As for the simple sketches, SketchRNN has a reasonable ability in completing the missing parts of each sketch. For example, we can observe the general good examples from the first column of SketchRNN in Fig. 3. However, SketchRNN is quite limited to fill the complicated sketches, such as the *flashlight,tiger*, SketchRNN may be failed to complete them. In contrast, our Sketch-BERT can still correctly capture both the shape and details of such sketches as the results in the second and third columns of Fig. 3. We also show more examples of different classes on sketch

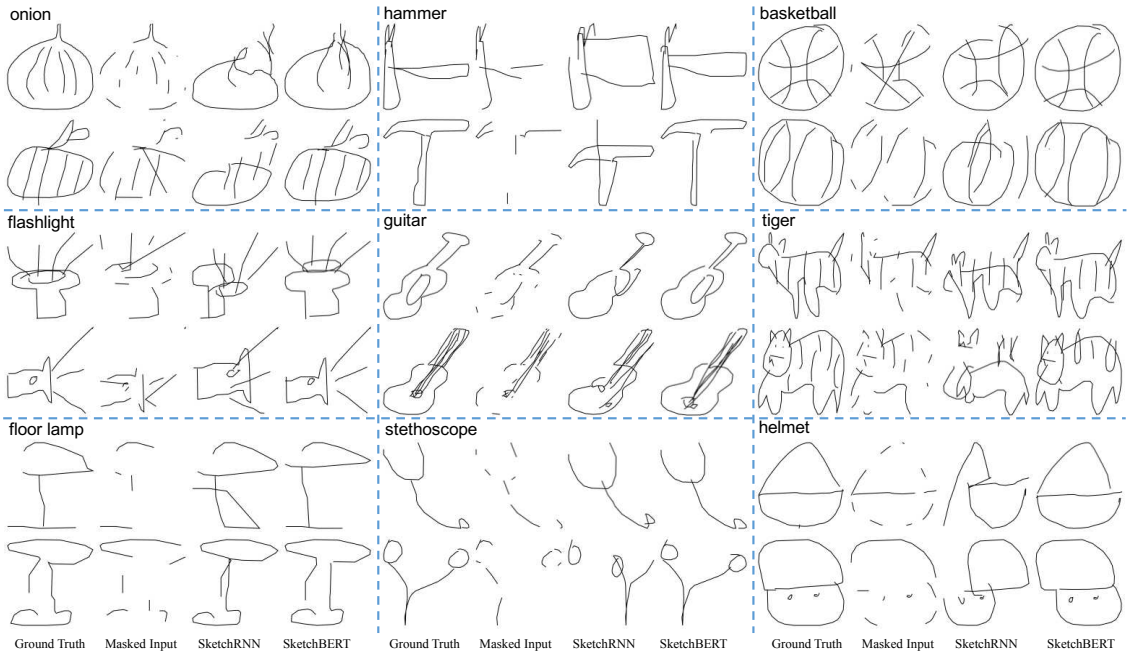


Figure 3. Completion results on sketch gestalt of our Sketch-BERT and SketchRNN on QuickDraw dataset from 9 classes, *onion,flashlight, floor lamp, hammer, guitar, stethoscope,basketball, tiger,helmet*.

Models	Classification		Retrieval	
	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5(%)
Single	86.51	96.72	81.73	92.13
Position	87.37	97.01	82.22	91.98
State	86.83	96.88	81.87	92.15
Full	**88.30**	**97.82**	**85.47**	**93.49**

Table 3. The performance of classification and retrieval tasks on QuickDraw dataset after different types of pre-training tasks.

gestalt task in supplementary material. Besides the qualitative results, we also provide a user study as the quantitative comparison in the supplementary material.

4.5. Pre-training Task Analysis

In this section, we give further ablation study and analyze how the self-supervised learning and models can affect the performance on sketch representation learning.

Different Pre-training Tasks. First, we study the different pre-training tasks in our model: (1) Single, means the traditional random mask strategy used in BERT; (2)Position, means that only masks the position information according to the mask strategy in our sketch gestalt model; (3)State, masks the state information, (4) Full, is the full newly proposed mask strategy in sketch gestalt model. We show the performance of standard Sketch-BERT on the classification and retrieval tasks after these pre-training tasks in Tab. 3.

It is clear that our sketch gestalt model plays an important role to improve the performance of Sketch-BERT,

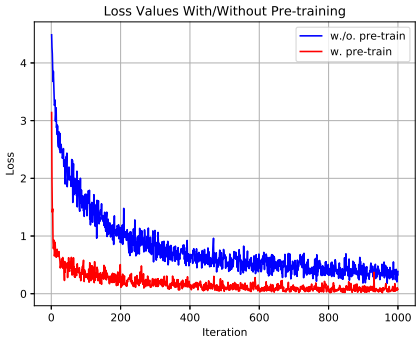


Figure 4. Convergence Rate with/without Pre-training of Sketch-BERT on QuickDraw dataset.

and we notice there is a consistent improvement over the other mask models: Single ($> 1.7\%$), Position ($> 1\%$), State ($> 1.4\%$). This reveals the significance of a proper mask model for learning the good neural representation of sketches. Furthermore, we can find the position information plays a more important role to sketch representation learning than the state information, as in Tab. 3.

Faster Convergence Rate of self-supervised learning by Sketch Gestalt Model. In addition to the improvement on classification, we also find that the pre-training sketch gestalt model can significantly reduce the training epochs for the convergence of classification task. As the curves shown in Fig. 4, the Sketch-BERT will converge much

Models	Classification (%)		Retrieval(%)	
	Top-1	Top-5	Top-1	Top-5
$345 \times 70K$	**88.30**	**97.82**	**85.47**	**93.49**
$345 \times 5K$	85.73	97.31	82.44	92.13
$200 \times 5K$	84.89	97.14	81.87	92.07
$100 \times 5K$	85.82	97.31	81.91	92.01

Table 4. The performance of classification and retrieval tasks of Sketch-BERT with different volumes of pre-training data.

Models	Classification		Retrieval	
	Top-1 (%)	Top-5 (%)	Top-1 (%)	Top-5(%)
6-8-256	84.83	96.42	81.06	91.86
12-8-256	86.34	97.15	83.23	92.13
12-16-1024	85.31	97.44	82.76	92.11
8-12-768	**88.30**	**97.82**	**85.47**	**93.49**

Table 5. The performance of classification and retrieval tasks for different structures of Sketch-BERT ($L - A - H$).

faster after pre-training on Quick-Draw dataset, from about 50 to 5 epochs where one epoch has 50 Iterations in Fig. 4.

Different Volumes of Pre-training Tasks. We also study how the volume of pre-training data affects the downstream tasks. We test the classification and retrieval tasks on 100 classes with 5K training, 2K validation and 2K test samples in QuickDraw dataset. By varying the number of classes and the number of training samples in each class, we get different settings for pre-training tasks as shown in Tab. 4. We denote the volume of pre-training data as $c \times n$, where c is the number of classes and n is the number of training samples in each class. We can find there is no obvious improvement after increasing the number of categories for pre-training data. But the number of pre-training samples in each class affects the performance in a more fundamental way, as reflected by the 3% improvement on top-1 accuracy.

Sketch-BERT Architecture Analysis. We further compare different variants of Sketch-BERT, as shown in Tab.5. We show that a reasonable depth and width of the network is important to Sketch-BERT. Particularly, We denote the structure of Sketch-BERT by three key hyper-parameters $L - A - H$: number of layers L, number of self-attention heads A, hidden size H. It shows that the architecture $8 - 12 - 768$ makes a good balance between the model complexity and final performance of Sketch-BERT model, if compared against the other variants. When hidden size is small,*e.g.*, $H = 256$, a deeper Sketch-BERT can help increase the capacity for learning representation of sketches, clarified by the 2% improvement from $L = 6$ to $L = 12$ on both classification and retrieval tasks. Nevertheless, we found the Sketch-BERT with 12 layers ($12-16-1024$) has slightly inferior results to the other variants, and hard to get converged.

Sketch Gestalt by CNN based Model. We further conduct experiment to show that the proposed sketch gestalt task is very difficult. We use the Gated Convolution [26] model to train on QuickDraw dataset with random masks. It is difficult for such CNN based model to reconstruct the shape of complicated sketches; and the results always exist artifacts. Since the different input requirement of image inpainting and sketch gestalt, the "Masked Input" terms in Fig. 5 use irregular masks which is fundamentally different from the terms in Fig. 3. The models for image inpainting always aim at recovering the masked parts by borrowing the patches from other parts of the image, while it is not tailored to sketch gestalt.

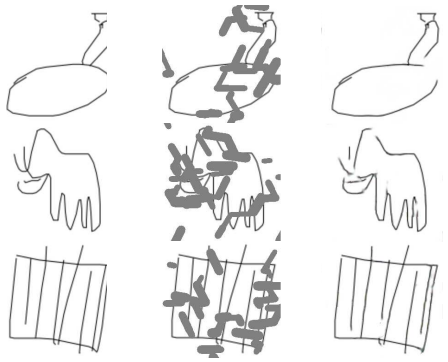


Figure 5. CNN based models for sketch gestalt Task. We employ Gated Convolution [26] to complete the masked sketches.

5. Conclusion

In this work, we design a novel Sketch-BERT model for sketch representation learning which employs the efficient self-supervised learning by sketch gestalt. A novel sketch gestalt model is proposed for self-supervised learning task of sketches. The results on QuickDraw and TU-Berlin datasets show the superiority of Sketch-BERT on classification and retrieval tasks. We also conduct experiments on sketch gestalt task to show the ability of Sketch-BERT on generative representation learning. Furthermore, the Sketch-BERT model can be extended to more tasks for sketches like sketch based image retrieval and sketch generation which can be studied in future.

6. Acknowledgements

This work was supported in part by NSFC Projects (U1611461,61702108), Science and Technology Commission of Shanghai Municipality Projects (19511120700), Shanghai Municipal Science and Technology Major Project (2018SHZDZX01), and Shanghai Research and Innovation Functional Program (17DZ2260900).

References

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014. 2

[2] Agne Desolneux, Lionel Moisan, and Jean-Michel Morel. Gestalt theory and computer vision. In *Theory and Decision Library A:*, 2004. 1

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. (document), 1, 2, 3.1, 3.2, 3.4

[4] Mathias Eitz, James Hays, and Marc Alexa. How do humans sketch objects? *SIGGRAPH*, 2012. 4.1

[5] Mathias Eitz, Kristian Hildebrand, Tamy Boubekeur, and Marc Alexa. Sketch-based image retrieval: Benchmark and bag-of-features descriptors. *TVCG*, 2010. 4.1

[6] S. Gidaris, P. Singh, and N. Komodakis. Unsupervised rep- resentation learning by predicting image rotations. In *ICLR*, 2018. 2

[7] David Ha and Douglas Eck. A neural representation of sketch drawings. In *ICLR*, 2018. (document), 1, 2, 3.1, 4.1, 4.1, 4.4

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 4.1, 4.1

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 2, 4.1, 4.1

[10] Rui Hu and John Collomosse. A performance evaluation of gradient field hog descriptor for sketch based image retrieval. *CVIU*, 2013. 2

[11] Rui Hu, Tinghuai Wang, and John Collomosse. A bag-of-regions approach to sketch-based image retrieval. In *ICIP*. IEEE, 2011. 2

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 2, 4.1

[13] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. *ICCV*, 2019. 2

[14] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, 2017. 2

[15] Alexander Kolesnikov, Xiaohua Zhai, and Lucas Beyer. Revisiting self-supervised visual representation learning. In *CVPR*, 2019. 2

[16] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019. 1, 3.2

[17] Lei Li, Changqing Zou, Youyi Zheng, Qingkun Su, Hongbo Fu, and Chiew-Lan Tai. Sketch-r2cnn: An attentive network for vector sketch recognition. *arXiv preprint arXiv:1811.08170*, 2018. 1, 2

[18] Yijun Li, Chen Fang, Aaron Hertzmann, Eli Shechtman, and Ming-Hsuan Yang. Im2pencil: Controllable pencil illustration from photographs. In *CVPR*, 2019. 2

[19] Yi Li, Yi-Zhe Song, and Shaogang Gong. Sketch recognition by ensemble matching of structured features. In *BMVC*, 2013. 2, 4.1, 4.1

[20] Hangyu Lin, Peng Lu, Yanwei Fu, Shaogang Gong, Xiangyang Xue, and Yu-Gang Jiang. Tc-net for isbir: Triplet classification network for instance-level sketch based image retrieval. In *ACM Multimedia*, 2019. 2, 3.4, 4.1, 4.1

[21] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash. Boosting self-supervised learning via knowledge transfer. In *CVPR*, 2018. 2

[22] Jifei Song, Qian Yu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017. 1, 2, 4.1, 4.1

[23] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeualPS*, 2017. 2

[24] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*, 2019. 2

[25] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 2

[26] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *ICCV*, pages 4471–4480, 2019. 2, 5, 4.5

[27] 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, 2016. 2, 4.1, 1, 4.1

[28] Qian Yu, Yongxin Yang, Feng Liu, Yi-Zhe Song, Tao Xiang, and Timothy M Hospedales. Sketch-a-net: A deep neural network that beats humans. *IJCV*, 122(3):411–425, 2017. 1, 2, 4.1

[29] R. Zhang, P. Isola, and A. A. Efros. Colorful image colorization. In *ECCV*, 2016. 2

[30] Xu-Yao Zhang, Fei Yin, Yan-Ming Zhang, Cheng-Lin Liu, and Yoshua Bengio. Drawing and recognizing chinese characters with recurrent neural network. *TPAMI*, 40(4):849–862, 2017. 2

[31] Tao Zhou, Chen Fang, Zhaowen Wang, Jimei Yang, Byungmoon Kim, Zhili Chen, Jonathan Brandt, and Demetri Terzopoulos. Learning to sketch with deep q networks and demonstrated strokes. *arXiv preprint arXiv:1810.05977*, 2018. 2

[32] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017. 2