Sketch-Segformer: Transformer-Based Segmentation for Figurative and Creative Sketches

Yixiao Zheng; Jiyang Xie; Aneeshan Sain; Yi-Zhen Song; Zhanyu Ma

摘要

草图是目前视觉学界研究得比较透彻的一个课题。尤其是草图语义分割，是实现更精细草图解读的基础步骤。最近的研究采用各种方法从草图中提取辨别特征，大大提高了分割的准确性。常见的方法包括关注草图图像的整体、笔画级表示或其中蕴含的序列信息。然而，这些方法大多只关注这些多方面信息的一部分。在本文中，本论文首次证明了在草图数据的所有这三个方面都有互补信息可供挖掘，而且由于对草图特定信息的挖掘，分割性能也会随之提高。具体来说，本论文提出了草图-分割器（Sketch-Segformer），这是一种基于Transformer的草图语义分割框架，本质上将草图视为笔画序列而非像素图。特别是，Sketch-Segformer 引入了两种类型的自注意力模块，它们具有类似的结构，可用于整个草图或单个笔画这两个不同的感受野。并且，顺序嵌入与从整个草图以及局部笔画级信息中学习到的空间嵌入进一步协同。

关键词：草图语义分割；二维结构化点集表示；双自注意力模块、阶次嵌入、基于草图特定Transformer的框架

1 引言

草图通常以三种不同的数据格式表示： 光栅图像、点序列和图或者点集。最近的草图分割方法[1][8][10][11][12]相应地采用了不同的框架：基于图像的方法[10][12][13]，基于序列的方法[1][5][14]和基于图的方法[8][11][15]。随着深度学习技术的出现[16][17][18][19][20]，现有的方法取得了很好的性能，但通常单独地探索草图中蕴含的多种类型的特征信息。基于图像的方法将光栅草图图像作为输入，从而忽略了草图的绘制顺序。由于基于序列的方法只考虑笔画点的相对坐标和笔的动作，因此通常不能很好地编码结构信息。基于图形的方法将草图视为图形或点集。它们在捕捉草图的空间信息和笔画信息方面具有很大的潜力。然而，基于图形的方法通常会忽略草图中包含的序列信息。需要注意的是，笔画序列仍然是解释和理解草图的有用信息[2][3][9]，在草图语义分割任务中不应被忽视。综上所述，学习有效草图分割网络的关键在于设计一个综合框架，同时考虑序列、全局和笔画级这三个方面的草图特定信息，如图 1(a)所示。

鉴于基于Transformer的方法在各种视觉和 NLP 任务中的广泛应用[21][22][23][24]，这种方法听起来可

能过于简单，但在获得良好的序列嵌入以考虑草图的绘制顺序方面，这种方法既直观又至关重要。为了容纳草图级和笔画级信息，在Transformer设置的基础上进一步引入草图级和笔画级这两种自注意力模块。草图级自注意力模块在整个草图上操作，以获取草图级结构信息，而笔画级自注意力模块则侧重于笔画形状信息，只在每个笔画内操作。

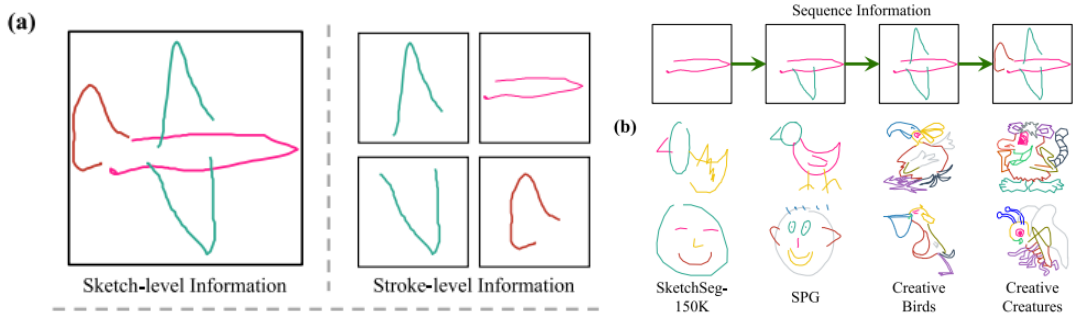

图 1 (a)草图级信息和笔画级信息的形式示意图。(b)一些数据集上的草图示例

2 相关工作

2.1 草图语义分割

与以往依赖手工特征和复杂模型的工作相比[7][25][26]，许多深度学习模型[8][10][11][12][14] 在草图语义分割任务上取得了很好的性能。根据现有方法使用的数据格式，本论文将它们大致分为三类，即基于图像的方法、基于序列的方法和基于图的方法。

1) 基于图像的方法： 与其他计算机视觉任务类似，研究人员通常将草图的光栅图像作为模型的输入。Zhu 等人[27]提出了一种双卷积神经网络（CNN）模型，其中一个 CNN 采用大尺寸卷积核来处理长草图，而另一个 CNN 则采用小尺寸核来处理短草图。Li 等人[10]提出了一种沙漏形网络，其中包含用于草图语义分割的编码器和解码器。 他们将三维模型中现有的分割和标签转移到手绘草图中，从而避免了将大量注释良好的草图作为训练数据的需要。 Zhu 等人[12]提出了一种基于 CNN 和条件随机场（CRF）的混合方法。然而，上述基于图像的方法通常会忽略草图的绘制顺序或笔画结构。相比之下，本文提出的 Sketch-Segformer 采用序列嵌入法对草图的绘制顺序进行编码，并使用笔画自注意力模块捕捉笔画内部的结构和上下文信息。

2) 基于序列的方法： 由于草图的时间性，大多数研究者使用关键点的相对坐标和笔的动作状态来表示输入的草图。Wu 等人[1][14]将草图语义分割任务视为序列到序列生成问题，提出了一种基于递归神经网络（RNN）的模型，将笔画点序列转换为语义标签序列。Li 等人[5]在其草图分组模型获得的特征表征上应用了额外的软最大激活全连接层（FC），并将草图的每个片段分类为语义标签。然而，上述基于序列的方法使用 RNN 模型来操作草图点的相对坐标，通常会忽

略不同笔画之间点的邻近性。 与这些基于序列的方法不同，本文提出的 Sketch-Segformer 使用笔画点的二维绝对坐标，并利用草图自注意力模块从全局视图中提取上下文信息。

3) 基于图形的方法：受点云分析的启发，研究界也将草图视为图或点集[8][11][15]。Wang 等人[11] 提出了多列点神经网络（MCPNet），它将草图的采样点作为输入，并采用多列不同大小的滤波器来更好地捕捉草图的结构。Yang等人[8]使用双分支图神经网络（GNN），采用静态和动态图卷积单元分别提取笔画内和笔画间的特征。图表示法和 GNN 模型适用于对草图和笔画结构进行编码，但现有的基于图的方法很少考虑草图的绘制顺序和序列信息。与 SketchGNN 中使用的图卷积操作不同，本文提出的 Sketch-Segformer 建立了双自注意力模块，专门针对草图的草图结构信息和笔画结构信息设计了两种类型的自注意力模块。借助序列嵌入模块，Sketch-Segformer 可以关注 SketchGNN 所忽略的草图绘制顺序。

2.2 Transformer

Transformer 除了在自然语言处理领域取得巨大成功外，[24] 还被应用于许多计算机视觉任务，如图像分类 [28][29]、物体检测 [30][31]、语义分割 [21][32] 和点云分析 [22][33]。许多研究人员致力于将自我注意机制应用于图像理解 [34][35][36]。为了解决草图研究的问题，Transformer也被推广到草图领域 [4][23][37][38]。Lin 等人[4]提出了一个从Transformer学习草图双向编码器表示的模型（Sketch-BERT）和一个帮助训练 Sketch-BERT 的草图格式塔模型（SGM）。Ribeiro 等人[23] 提出了用于自由手绘草图的 Sketchformer，并开发了几种变体来处理连续和标记化形式的草图序列。Aksan 等人提出的 Cose [39]，通过忽略单个笔画的排序，解决了因数据的组成性质而引起的复杂性问题。它使用基于Transformer的关系模型来更好地捕捉生成任务中笔画之间的关系。

Bhunia 等人[38]提出了一个从粗到细的两阶段框架，该框架将草图生成问题分解为创建粗草图组合，然后在草图中加入精细细节。基于变压器的网络在草图识别、草图检索、草图生成和其他与草图相关的任务中取得的成功表明，它可以处理草图。据本论文所知，所提出的草图分割器是第一个专门为草图语义分割而设计的基于Transformer的网络。

3 本文方法

3.1 数据表示

本文工作为了能够更好地同时协同到全局、笔画级和序列信息这三类特征，将草图转化为用n个2D点构成的点击来表示草图数据 $\mathbf{X}_i = (P_i, E_i, R_i)$ 表示第i个草图:

$P_i = \left\{\left(x_{i,j}, y_{i,j}\right)\right\}_{j=1}^{N}$ 表示第 i 个草图中的 N 个点。其中 $x_{i,j}$ 和$y_{i,j}$ 表示的是第j个点 $p_{i,j}$ 的2D绝对坐标位置。

$E_i = \{(e_{i,k})\}_{k=1}^{M}$ 表示第 i 个草图中的 M 个笔画情况，其中 $e_{i,k} = \left\{p_{i,\bar{j}}\right\}$ 表示了属于第 k 个笔画的所有点的集合。

$R_i = \{(r_{i,j})\}_{j=1}^{N}$ 表示人工为每个点标注的标签

3.2 草图级注意力模块

自注意力模块是 Transformer[24]全局语境信息学习模型的关键组成部分。首先，自注意力模块将输入的每个元素投射为三个向量，即 query、key 和 value。任何两个元素之间的注意力权重都是通过计算它们的 query 和 key 向量的点积得到的。每个元素的输出特性由所有值向量与注意力权重的加权和生成，这使得每个元素的输出特性与所有输入特征相关。

为了有效提取全局和笔画级别的语义上下文信息，本论文使用了两种类型的自注意力模块，即草图和笔画自注意力模块。从形式上看，给定一个 N 点草图，其点 d_e 维嵌入特征 $\mathbf{F}_e \in \mathbb{R}^{N\times d_e}$，草图向自注意力模块首先将这些特征转化为 query、key 和 value 矩阵（分别表示为 $\mathbf{Q} \in \mathbb{R}^{N\times d_{qk}}$, $\mathbf{K} \in \mathbb{R}^{N\times d_{qk}}$ 和 $\mathbf{V} \in \mathbb{R}^{N\times d_v}$，如下所示：

$$(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{F}_e \cdot \left(\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v\right), \tag{1}$$

其中 $W_q \in \mathbb{R}^{d_e\times d_{qk}}$，$W_k \in \mathbb{R}^{d_e\times d_{qk}}$，和 $W_v \in \mathbb{R}^{d_e\times d_v}$ 是可学习的参数矩阵。为了提高计算效率，本论文将 d_{qk} 设为 $d_e/4$。然后，本论文通过 query 向量和 key 向量的归一化点积计算注意力权重$\widehat{\mathbf{A}} = \{\alpha_{j,\hat{j}}\} \in \mathbb{R}^{N\times N}$，如下所示：

$$\begin{aligned} \hat{\mathbf{A}} = \{\hat{\alpha}_{j,\hat{j}}\} &= \frac{\mathbf{Q}\cdot\mathbf{K}^T}{\sqrt{d_{qk}}}, \\ \alpha_{j,\hat{j}} = \mathrm{softmax}\left(\hat{\alpha}_{j,\hat{j}}\right) &= \frac{\exp\left(\hat{\alpha}_{j,\hat{j}}\right)}{\sum_{\bar{j}=1}^{N}\exp\left(\hat{\alpha}_{j,j}\right)}. \end{aligned} \tag{2}$$

草图自注意力模块的输出特性 $\mathbf{F}_{GA} \in \mathbb{R}^{N\times d_v}$ 是由value向量与相应的注意权重加权求和得到的，如下所示：

$$\mathbf{F}_{GA} = \mathbf{A}\cdot\mathbf{V} \tag{3}$$

通过对属于草图的所有点进行自注意操作，草图自注意力模块旨在能够捕捉全局上下文信息。

3.3 笔画级注意力机制

笔画自注意力模块的结构与草图自注意力模块类似，但工作的感受野不同。具体来说，笔画级自注意力模块只计算对应笔画内每个点的关注权重和值向量的加权和。在使用公式 1 进行转换后（$\mathbf{W}_q, \mathbf{W}_k$ 和 $\mathbf{W}_v$ 参数在这两种自注意力模块之间不共享），第 k 个笔画 e_k 中第 j 个点 p_j 的输入特征被转换成相应的 query 向量、key 向量和 value 向量（分别表示为 $q_j \in \mathbb{R}^{d_{qk}}, k_j \in \mathbb{R}^{d_{qk}}$ 和 $v_j \in \mathbb{R}^{d_v}$）。然后，笔画自注意力模块只计算第 k 个笔画 e_k 中的点 p_j 的关注权重，计算公式如下：

$$\begin{aligned} \hat{\alpha}_{j,\hat{j}} &= \frac{q_j \cdot k_{\hat{j}}}{\sqrt{d_{qk}}},\ p_{\hat{j}} \in e_k, \\ \alpha_{j,\hat{j}} &= softmax\left(\hat{\alpha}_{j,\hat{j}}\right) = \frac{exp\left(\hat{\alpha}_{j,\hat{j}}\right)}{\sum_{\forall p_{\bar{j}} \in e_k} exp\left(\hat{\alpha}_{j,\bar{j}}\right)}. \end{aligned} \tag{4}$$

由笔画自注意力模块生成的点 p_j 的输出特征 $f_j \in \mathbb{R}^{d_v})$ 由第 k 个笔画 e_k 内的值向量加权求和得到：

$$f_j = \sum_{\forall p_{\bar{j}} \in e_k} \alpha_{j,\bar{j}} v_{\bar{j}}. \tag{5}$$

换句话说，输出 $\mathbf{F}_{SA} = \{f_j\}_{j=1}^{N} \in \mathbb{R}^{N \times d_v}$ 可视为仅在每个笔划内操作权重共享草图级别自注意力模块所获得的结果。由于感受野不同于草图级别自注意力模块，笔画级别自注意力模块更侧重于笔画结构和笔画上下文信息。

3.4 双分支注意力机制

有了上面介绍的两种自注意力模块，本论文为草图分割器设计了双重自注意力模块（如图 2 所示），以同时有效地捕捉草图的草图信息和笔画信息。从形式上看，给定一个 N 点草图，其点 d_e 维嵌入特征 $\mathbf{F}_e \in \mathbb{R}^{N \times d_e}$ 和 M 个笔画 $E = \{e_k\}_{k=1}^{M}$ 时，双自注意力模块首先并行操作草图自注意力模块(GA(·))和笔画自注意模(SA(·)),具体如下：

$$\begin{aligned} \mathbf{F}_{GA} &= \mathrm{BN}\big(\mathrm{GA}(\mathbf{F}_e)\big) + \mathrm{LB}(\mathbf{F}_e), \\ \mathbf{F}_{SA} &= \mathrm{BN}(\mathrm{SA}(\mathbf{F}_e, E)) + \mathrm{LB}(\mathbf{F}_e), \end{aligned} \tag{6}$$

其中 BN 表示批量归一化操作[54]，LB 表示线性层后的批量归一化操作。受文献[18]、[30]和[45]的启发，本论文在双自注意区块中添加了残差连接，以增加信息流并促进特征学习。由于 $\mathbf{F}_{GA}, \mathbf{F}_{SA} \in \mathbb{R}^{N \times d_v}$，$\mathbf{F}_e \in \mathbb{R}^{N \times d_e}$, 本论文通过一个 LB 层添加残差池化连接，如图 2 左侧部分和公式

6 所示。

运行这两类自注意力模块后，通过对 $\mathbf{F}_{GA}$ 和 $\mathbf{F}_{SA}$ 的串联进行残差变换得到双自注意力模块的输出特性 $\mathbf{F}_{DA} \in \mathbb{R}^{N\times d_e}$ ，如下所示：

$$\begin{aligned}\hat{F}_{DA} &= \mathrm{Concat}(\mathbf{F}_{GA}, \mathbf{F}_{SA}),\\ \mathbf{F}_{DA} &= \mathrm{ReLU}\big(\mathrm{MLP}(\hat{F}_{DA}) + \hat{F}_{DA}\big),\end{aligned} \tag{7}$$

其中，Concat (·) 表示连接操作，MLP 表示多层感知器。借助草图自注意力模块和笔画自注意力模块，双自注意力模块的输出特性 $\mathbf{F}_{DA}$ 旨在提取和融合草图的草图结构信息和笔画结构信息。

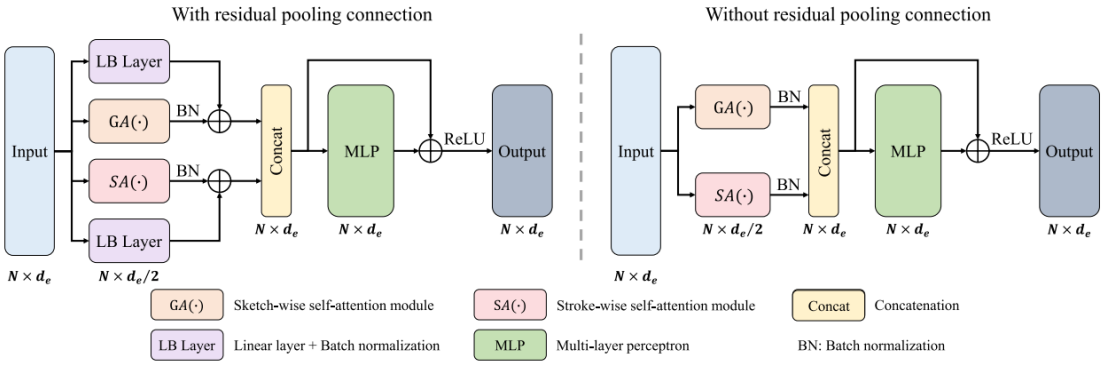

图 2 双分支自注意力模块示意图

3.5 模型总体框架

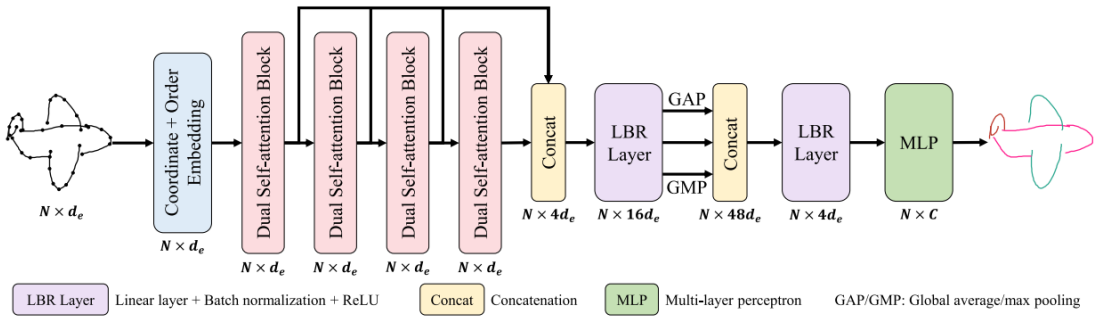

图 3 模型总体框架图

将所有组件整合在一起，本论文绘制了 Sketch-Segformer 的完整架构，如图 3 所示。Sketch-Segformer 的目的是将输入的草图点投影到更高维度的特征表示中，这些特征表示可以表征草图的序列信息、草图和笔画结构信息，从而完成草图语义分割任务。具体来说，Sketch-Segformer 使用可训练的顺序嵌入 $\mathbf{O} \in \mathbb{R}^{N\times d_e}$ 来编码顺序信息，即 $\mathbf{O}$ 的第 j 行编码第 j 个点的顺序信息，同时它堆叠了四个双重自我注意块来学习丰富的草图结构信息和笔画结构信息。形式上，给定一个 N

点草图，其坐标为 $\mathbf{P}=\{p_j\}=\{(x_j,y_j)\}_{j=1}^{N}\in\mathbb{R}^{N\times 2}$，其 M 个笔画 $E=\{e_k\}_{k=1}^{M}$，拟议的 Sketch-Segformer 首先将输入坐标嵌入到一个新的特征空间，然后在坐标嵌入中加入阶次嵌入，形成 d_e 维嵌入特征 $\mathbf{F}_e\in\mathbb{R}^{N\times d_e}$，如下所示：

$$\mathbf{F}_e=\mathrm{MLP}(\mathbf{P})+\mathbf{O}. \tag{8}$$

然后将嵌入特征 $\mathbf{F}_e$ 输入四个堆叠的双自注意力模块，如下所示：

$$\begin{aligned}&\mathbf{F}_1=DA^1(\mathbf{F}_e,E),\\&\mathbf{F}_l=DA^l(\mathbf{F}_{l-1},E),\ l=2,3,4,\end{aligned} \tag{9}$$

其中 DA^l 表示第 l 个双自注意力模块，$\mathbf{F}_l\in\mathbb{R}^{N\times d_e}$ 表示相应的输出特性。

为了形成最终 MLP 层的有效特征，本论文选择串联每个双自注意力模块的输出，并对学习到的点向特征表示进行全局平均和最大池化运算，如下所示：

$$\begin{aligned}\hat{\mathbf{F}}_o&=\mathrm{LBR}(\mathrm{Concat}(\mathbf{F}_1,\mathbf{F}_2,\mathbf{F}_3,\mathbf{F}_4)),\\\mathbf{F}_{GAP}&=\mathrm{Rep}\big(\mathrm{GAP}(\hat{\mathbf{F}}_o),N\big),\\\mathbf{F}_{GMP}&=\mathrm{Rep}\big(\mathrm{GMP}(\hat{\mathbf{F}}_o),N\big),\\\mathbf{F}_o&=\mathrm{LBR}\big(\mathrm{Concat}(\hat{\mathbf{F}}_o,\mathbf{F}_{GAP},\mathbf{F}_{GMP})\big),\end{aligned} \tag{10}$$

其中，LBR 表示线性层，之后是批量归一化操作和 ReLU 激活函数。

GAP 和 GMP 分别表示点式全局平均和最大池化操作。$Rep(\cdot,N)$ 表示重复输入 N 次的操作。最后，建议的草图分割器使用多层感知器（MLP）预测输入草图的最终按点分割结果ˆR，如下所示：

$$\hat{R}=\mathrm{MLP}\ (\mathbf{F}_o).$$

在训练期间，本论文使用交叉熵损失 $\mathcal{L}_{\mathrm{CE}}$ 对所提出的草图-分割器进行端到端优化，如下所示：

$$\min_{\theta}\mathop{\mathbb{E}}_{\mathbf{x}_i=(P_i,E_i,R_i)\sim\mathcal{D}_{\mathrm{train}}}\big[\mathcal{L}_{\mathrm{CE}}\big(\hat{R}_i,R_i;\theta\big)\big], \tag{11}$$

其中 $\mathcal{D}_{\mathrm{train}}$ 表示训练集，θ 表示所提出的模型的参数。

4 复现细节

4.1 与已有开源代码对比

由于本次实验复现的论文已开源，我们结合论文的基本框架和计算公式，参考开源代码进

行复现。复现的代码整体框架与开源代码相似，主要的复现工作为：

1）更改使用的数据集类：论文中提到，模型使用到的数据都是只由256个点构成的草图。基于此，项目开源代码中使用的是SketchGNN这个项目中提供的SPG256这个数据集，其中的草图数据都已经被处理成了由256个构成，并且已经对每个点都进行了语义标注。论文中作者说明了如何根据SPG这个草图中点数量不统一的原始数据集创建所有草图均由256个点表示的SPG256数据集的流程。由此，复现时首先根据论文中对原始数据集的处理流程，使用python完成新数据集创建的代码实现，然后再对新数据集中的数据进行预处理为模型所需的数据类型。 复现时创建的数据集和SPG256数据集类似，一共包含了20个类别，每个类别中有650个训练样本，100个测试样本，50个验证样本。所有的草图数据都只由256个点表示，包含的笔画数都在10以内。对于每个类别的草图数据，需要对模型进行分开训练。

2）更改模型结构：论文开源的项目代码中的模型结构与论文中给的模型框架描述有一些出入，复现时将严格按照论文中的模型框架图，使用pytorch完成模型的创建。

3）替换笔画级别的自注意力模块的实现方法：由论文可知，在计算笔画级别的自注意力模块时，只需要计算每个壁画内部的点之间的相关性即可。虽然每个草图经过数据预处理之后都已经被统一成由256个点，但是草图的笔画长度却是不统一的。因此，需要使用一种数据结构来表示每个笔画内部的点之间的连接关系，使得能够独立得对每条笔画进行自注意力计算操作。在论文开源的代码中，作者使用的方式是为每个笔画内部的点创建相应的图数据，其中顶点为草图中所有的点相应数据，如果两个点属于相同的笔画则创建一条有向边。在计算笔画级自注意力时借助图卷积神经网络的信息传递机制，引入这种点之间的连接关系即可。本人在进行复现的时选择了一种更为简单的方式来达到独立处理每个笔画的目的。首先在数据预处理时，创建一个参数来记录每个点所属的笔画的序号。借助这个参数，可以为要处理的笔画创建掩码矩阵，消除其他点对于该笔画的注意力计算时的影响。具体的实现思路类似于Transformer中padding mask机制，将本次不需要处理的点看作是padding的值，创建的掩码的相应位置的赋予一个很小的数值。在计算自注意力的过程中，将进行softmax之前的处理结果加上这个掩码矩阵的结果再进行softmax操作，从而可以使得不需要考虑的点的概率值为0，相当于只计算了所需笔画的内部点的自注意力，与论文中提及的笔画级自注意力模块所需要实现的功能一致，且实现更为简单。

4）通过实验对比复现的模型和论文作者提供的模型的效果。

本次论文复现重点过程在于论文中闪光点的学习吸收和项目中已有代码的关于闪光点的思路借鉴。

4.2 实验环境

Cuda = 10.2

Python = 3.6.13

Pytorch = 1.10.1

pytorch_geometric = 2.0.3

4.3 创新点

1） 创建了一种新的草图表示结构：将草图表示为二维结构化的点集，使用的是每个点的绝对坐标。

2） 双分支自注意力模块：分别针对草图的顺写法和顺笔画法结构信息设计了自注意模块，同时采用顺序嵌入对草图中包含的序列信息进行编码。

3） 使用了掩码机制实现笔画级自注意力的计算。

4） 重新生成了全部草图均只由256个点构成的数据集。

5 实验结果分析

本次实验使用的模型评价指标主要为1）p_metric: 衡量的是草图中所有点预测的准确率，使用所有预测正确的点的数量除以所有点的数量的方式计算所得；2）c_metric：衡量的是草图所有笔画的预测正确率，认为笔画内点的预测正确率大于等于75%的笔画是被正确预测的，通过所有预测正确的笔画的数量除以所有笔画的数量的方式计算得到c_metric的值；3）loss_avg: 所有草图预测的平均损失。

1） 两种笔画级自注意力实现方法对比实验

本次实验使用到的数据集均使用项目源码中使用的已经与处理好的数据集SPG256，同时两个模型除了笔画级别自注意力模块不同，其他均保持相同的结构。将两个模型在数据集中20个类别的草图上分别进行100个epoch的训练之后，得到了如图的实验结果。图4展示了airplane类别的验证集中所有草图的语义分割标注信息可视化结果，不同的颜色代表不同的语义标签值。图5表示的是由使用图结构实现笔画级自注意力模块的模型对验证集中airplane类别的所有草图进行语义分割的结果，即使用

论文作者提出的模型的结果。相应的，图6就是使用掩码机制实现笔画级自注意力模块的模型对验证集中airplane类别的所有草图进行语义分割的结果，即修改之后的模型的结果。表1中展示了两个模型在每个类别上进行语义分割的表现情况。

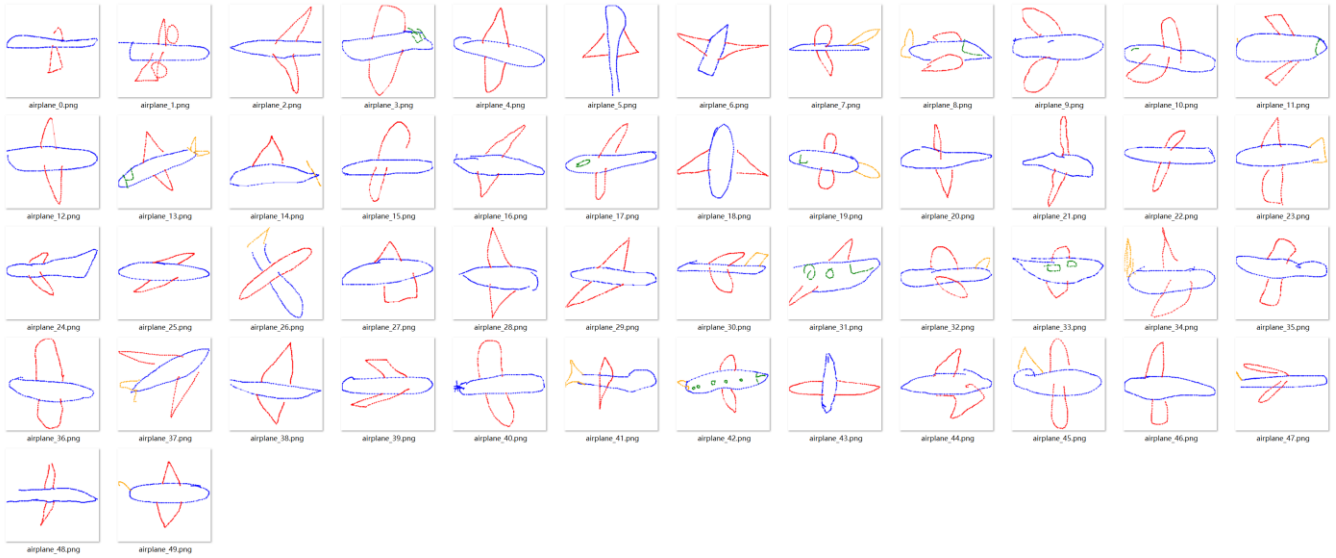

图 4 验证集airplane类别所有草图语义分割可视化

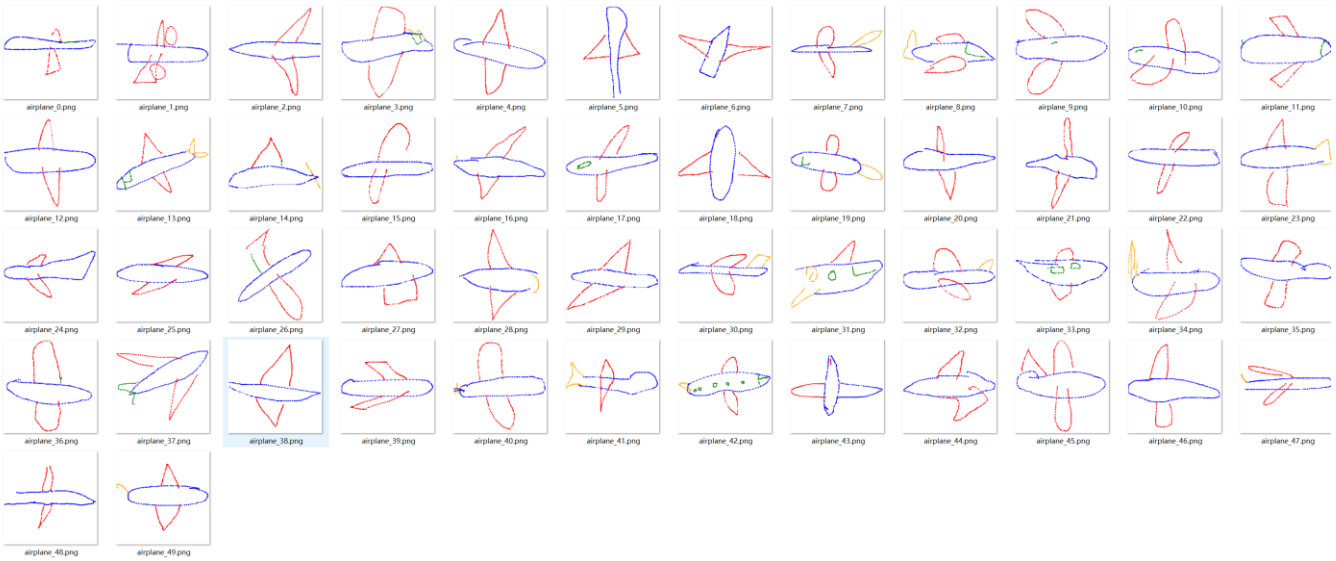

图 5 图结构实现笔画级自注意力块的模型语义分割结果可视化

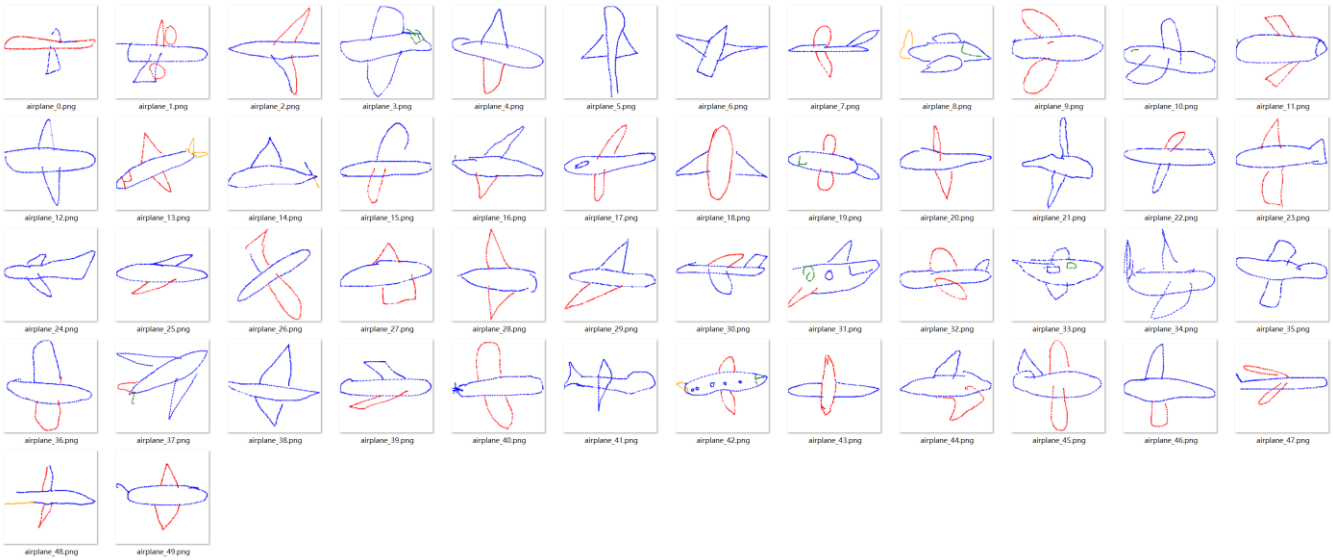

图 6 掩码方式实现笔画级自注意力块的模型语义分割结果可视化

表 1 论文模型和复现模型在SPG256数据集上的定量指标

Category	Sketch-SgeFormer			My-model		
	P_metric ↑	C_metric ↑	Loss_avg ↓	P_metric ↑	C_metric ↑	Loss_avg ↓
Airplane	95.98	91.56	0.011	87.37	79.87	0.023
Alarm clock	97.93	94.90	0.009	91.91	85.45	0.018
Ambulance	92.86	87.73	0.015	85.45	79.12	0.029
Ant	92.93	90.45	0.007	82.06	77.17	0.050
Apple	97.26	92.90	0.012	94.88	85.13	0.027
Backpack	91.10	83.86	0.010	83.54	73.15	0.029
Basket	96.12	95.67	0.002	92.40	90.80	0.030
Butterfly	98.84	97.03	0.006	95.11	88.60	0.016
Cactus	97.06	94.80	0.005	91.93	86.86	0.020
Calculator	99.16	98.19	0.005	97.22	92.92	0.009
Campfire	96.95	95.65	0.023	93.14	90.90	0.033
candle	98.83	97.30	0.004	96.58	91.18	0.033
Coffee cup	93.50	89.49	0.003	88.21	81.33	0.045
Crab	98.78	97.58	0.007	95.74	93.79	0.022
Duck	97.81	95.57	0.017	93.17	86.69	0.032
Face	98.65	97.05	0.002	94.67	89.31	0.026
Ice cream	94.23	92.96	0.002	89.81	86.95	0.018
Pig	98.72	97.63	0.006	90.49	83.24	0.024
Pineapple	98.54	95.02	0.004	96.42	93.36	0.022
Suitcase	99.64	98.05	0.008	96.38	93.02	0.011
Average	96.74	94.17	0.008	91.82	86.44	0.026

通过将图4分别与图5，图6进行对比可知，论文作者采用的笔画级自注意力计算策略的语义分结果更加接近原始数据集中的标注结果，并且从表1中可以发现，论文的原始方法使得模型在所有类别上的语义分割效果都远远优于使用掩码的方式。分析造成这种差距的原因应该是，在表示笔画内部点之间的关系时使用图的数据结构和图卷积神经网络的消息传递机制能够让模型学到更好的笔画内部点

之间的特征关系，从而使得模型有更好的语义分割效果。使用掩码机制的是实现方式虽然在结构和代码实现上更为简单，但是学到的特征也更少，因此进行语义分割的效果也有很大的提升空间。

2） 使用不同数据集对比实验

将论文提出的这个效果更好的模型同时在新创建的数据集和SPG256数据集中的airplane类别中进行100个epoch的训练，得到如下的实现结果。表2展示了模型在两个数据中的airplane类别上的表现情况。

表 2 论文模型在两个数据集上的定量指标

Dataset (airplane类)	Sketch-SgeFormer P_metric ↑	C_metric ↑	Loss_avg ↓
My-SPG256	61.69	43.33	0.192
SPG256	95.98	91.56	0.011

表2中的相应评价指标直观地体现出了模型在两个数据集上的效果差距。分析原因可能是在创建新数据集的这个过程中涉及到草图简化步骤，它需要使用到一个决定数据点是否需要保留的阈值。这个阈值参数的具体取值会影响到后续进行插值生成256个点的插值程度。为了验证阈值对模型效果的影响，我分别将阈值设为了0.08，0.5和1.0进行实验，结果如表3所示。可以发现，阈值的设置确实能够影响模型的效果，但是目前这三个选定的阈值仍然无法使得模型效果有很大的提升。

表 3 论文模型在不同阈值生成的数据集上的定量指标

Epsilon	Sketch-SgeFormer P_metric ↑	C_metric ↑	Loss_avg ↓
0.08	61.84	43.33	0.186
0.5	63.34	44.39	0.179
1.0	62.37	43.45	0.191

6 总结与展望

本文对论文《草图分割器：基于转换器的具象和创意草图分割》进行了复现，并对源代码进行了修改。论文的关键部分就是双分支自注意力模块，在对整个草图数据进行特征处理的同时还对每个笔画内部的特征进行学习，丰富了模型所学到的特征类型，并且这种联合了多种方面的特征十分适用于草图语义分割任务。

本人对双分支中的笔画级自注意力模块的实现方法进行了修改，使用更加简单直接的掩码机制来替代源代码中使用的笔画关联图结构。实验结果表明论文作者使用到的图卷积神经网络的消息传递机

制能够学到更好的笔画内部点之间的特征关系，从而使得模型的进行语义分割的效果更好。使用掩码机制的是实现方式虽然在结构和代码实现上更为简单，但是学到的特征也更少，因此进行语义分割的效果也有很大的提升空间。同时，本次复现还对数据预处理部分进行了修改，根据论文中的数据预处理流程，先将点数量不统一并且没有划分训练集、测试集、验证集的原始数据集SPG中的所有草图处理成为均由256个点表示的格式，同时划分出650个样本作为训练集、100个样本作为测试集和50个样本作为验证集。在这个过程中，进行草图简化的过程中需要使用到一个决定数据点是否需要保留的阈值，它的取值需要一定的经验，这也导致使用本人生成的数据集的语义分割效果不好。

通过本次复现，我深刻体会到草图语义分割仍然是一项具有挑战性的任务，需要进一步改进。因此，在未来，可以继续探索更好的方法来融合和建立草图和笔画上下文信息之间的交互，而不是简单的连接。

7 参考文献

[1] Y. Qi and Z.-H. Tan, “SketchSegNet+: An end-to-end learning of RNN for multi-class sketch semantic segmentation,” IEEE Access, vol. 7, pp. 102717–102726, 2019.

[2] S. Ge, V. Goswami, L. Zitnick, and D. Parikh, “Creative sketch generation,” in Proc. Int. Conf. Learn. Representations (ICLR), 2020, pp. 1–26.

[3] M. Eitz, J. Hays, and M. Alexa, “How do humans sketch objects?” ACM Trans. Graph., vol. 31, no. 4, pp. 1–10, Aug. 2012.

[4] H. Lin, Y. Fu, X. Xue, and Y.-G. Jiang, “Sketch-BERT: Learning sketch bidirectional encoder representation from transformers by selfsupervised learning of sketch gestalt,” in Proc. IEEE/CVF Conf. Comput.Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6757–6766.

[5] K. Li, K. Pang, Y.-Z. Song, T. Xiang, T. M. Hospedales, and H. Zhang, “Toward deep universal sketch perceptual grouper,” IEEE Trans. Image Process., vol. 28, no. 7, pp. 3219–3231, Jul. 2019.

[6] Z. Huang, H. Fu, and R. W. H. Lau, “Data-driven segmentation and labeling of freehand sketches,” ACM Trans. Graph., vol. 33, no. 6, pp. 1–10, Nov. 2014.

[7] R. G. Schneider and T. Tuytelaars, “Example-based sketch segmentation and labeling using CRFs,” ACM Trans. Graph., vol. 35, no. 5, pp. 1–9, Sep. 2016.

[8] L. Yang, J. Zhuang, H. Fu, X. Wei, K. Zhou, and Y. Zheng, “SketchGNN: Semantic sketch segmentation with graph neural networks,” ACM Trans. Graph., vol. 40, no. 3, pp. 1–13, Jun. 2021.

[9] K. Mukherjee, R. X. Hawkins, and J. W. Fan, “Communicating semantic part information in drawings,” in Proc. Annu. Conf. Cognit. Sci. Soc.(CogSci), 2019, pp. 1–7.

[10] L. Li, H. Fu, and C.-L. Tai, “Fast sketch segmentation and labeling with deep learning,” IEEE Comput. Graph. Appl., vol. 39, no. 2, pp. 38–51, Mar. 2019.

[11] F. Wang et al, “Multi-column point-CNN for sketch segmentation,” Neurocomputing, vol. 392, pp. 50–59, Jun. 2020.

[12] X. Zhu, Y. Xiao, and Y. Zheng, “2D freehand sketch labeling using CNN and CRF,” Multimedia Tools Appl., vol. 79, nos. 1–2, pp. 1585–1602, Jan. 2020.

[13] X. Zhu, J. Yuan, Y. Xiao, Y. Zheng, and Z. Qin, “Stroke classification for sketch segmentation by fine-tuning a developmental VGGNet16,” Multimedia Tools Appl., vol. 79, nos. 45–46, pp. 33891–33906, Dec. 2020.

[14] X. Wu, Y. Qi, J. Liu, and J. Yang, “SketchSegNet: A RNN model for labeling sketch strokes,” in Proc. IEEE Int. Workshop Mach. Learn.Signal Process. (MLSP), 2018, pp. 1–6.

[15] Y. Zheng, J. Xie, A. Sain, Z. Ma, Y. Z. Song, and J. Guo, “ENDE-GNN: An encoder–decoder GNN framework for sketch semantic segmentation,” in Proc. VCIP, 2022, pp. 1–5.

[16] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, pp. 436–444, May 2015.

[17] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR), Jun. 2016, pp. 770–778.

[18] D. Chang, K. Pang, Y. Zheng, Z. Ma, Y.-Z. Song, and J. Guo, "Your 'flamingo' is my 'bird': Fine-grained, or not," in Proc. IEEE/CVF Conf.Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 11471–11480.
[19] J. Xie, Z. Ma, D. Chang, G. Zhang, and J. Guo, "GPCA: A probabilistic framework for Gaussian process embedded channel attention," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 11, pp. 8230–8248, Nov. 2022.
[20] J. Xie et al, "Advanced dropout: A model-free methodology for Bayesian dropout optimization," IEEE Trans. Pattern Anal. Mach. Intell., vol. 44, no. 9, pp. 4605–4625, Sep. 2022.
[21] Z. Liu et al, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021, arXiv:2103.14030.
[22] M.-H. Guo, J.-X. Cai, Z.-N. Liu, T.-J. Mu, R. R. Martin, and S.-M. Hu, "PCT: Point cloud transformer," Comput. Vis. Media, vol. 7, no. 2, pp. 187–199, Jun. 2021.
[23] L. S. F. Ribeiro, T. Bui, J. Collomosse, and M. Ponti, "Sketchformer: Transformer-based representation for sketched structure," in Proc.IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 14141–14150.
[24] A.Vaswani et al, "Attention is all you need," in Proc. Adv. Neural Inf.Process. Syst. (NeurIPS), 2017, pp. 1–12.
[25] A. Delaye and K. Lee, "A flexible framework for online document segmentation by pairwise stroke distance learning," Pattern Recognit., vol. 48, no. 4, pp. 1197–1210, Apr. 2015.
[26] Z. Sun, C. Wang, L. Zhang, and L. Zhang, "Free hand-drawn sketch segmentation," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2012, pp. 626–639.
[27] X. Zhu, Y. Xiao, and Y. Zheng, "Part-level sketch segmentation and labeling using dual-CNN," in Int. Conf. Neural Inf. Process. (ICONIP), 2018, pp. 374–384.
[28] A. Dosovitskiy et al, "An image is worth 16×16 words: Transformers for image recognition at scale," in Proc. Int. Conf. Learn. Represent.(ICLR), 2020, pp. 1–12.
[29] H. Fan et al, "Multiscale vision transformers," in Proc. IEEE/CVF Int.Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 6804–6815.
[30] N. Carion et al, "End-to-end object detection with transformers," in Proc. Eur. Conf. Comput. Vis. (ECCV), 2020, pp. 213–229.
[31] Y. Li et al, "MViTv2: Improved multiscale vision transformers for classification and detection," 2021, arXiv:2112.01526.
[32] B. Wu et al, "Visual transformers: Token-based image representation and processing for computer vision," 2020, arXiv:2006.03677.
[33] L. Tang, Y. Zhan, Z. Chen, B. Yu, and D. Tao, "Contrastive boundary learning for point cloud segmentation," in Proc. IEEE/CVF Conf.Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 8479–8489.
[34] P. Ramachandran, N. Parmar, A. Vaswani, I. Bello, A. Levskaya, and J. Shlens, "Stand-alone self-attention in vision models," in Proc. AdvNeural Inf. Process. Syst. (NeurIPS), vol. 32, 2019, pp. 1–13.
[35] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.(CVPR), Jun. 2020, pp. 10073–10082.
[36] C. Doersch, A. Gupta, and A. Zisserman, "CrossTransformers: Spatiallyaware few-shot transfer," in Proc. Adv. Neural Inf. Process. Syst.(NeurIPS), 2020, pp. 1–13.
[37] A. S. Parihar, G. Jain, S. Chopra, and S. Chopra, "SketchFormer: Transformer-based approach for sketch recognition using vector images," Multimedia Tools Appl., vol. 80, no. 6, pp. 9075–9091, Mar. 2021.
[38] A. Kumar Bhunia et al, "DoodleFormer: Creative sketch drawing with transformers," 2021, arXiv:2112.03258.
[39] E. Aksan, T. Deselaers, A. Tagliasacchi, and O. Hilliges, "CoSE: Compositional stroke embeddings," in Proc. Adv. Neural Inf. Process.Syst. (NeurIPS), 2020, pp. 1–12.