Component-Level Segmentation for Oracle Bone Inscription Decipherment

Zhikai Hu[1], Yiu-ming Cheung[1]*, Yonggang Zhang[1], Zhang Peiying[2], Tang Pui Ling[2]

[1] Department of Computer Science, Hong Kong Baptist University, Hong Kong SAR, China
[2] Schoole of Chinese, The University of Hong Kong, Hong Kong SAR, China
{cszkhu, ymc, csygzhang}@comp.hkbu.edu.hk, {zhangpy, tangpl}@hku.hk

Abstract

Oracle Bone Inscriptions (OBIs), as the earliest systematically organized pictographic script in China, hold significant importance in the study of the origins of Chinese civilization. Of the approximately 4,500 excavated OBI characters, only about one-third have been deciphered, leaving the remaining characters shrouded in mystery. Over the past decade, an increasing number of researchers have attempted to leverage artificial intelligence to assist in deciphering OBIs, but these efforts have not yet fully met the demands of this challenging objective. In this paper, we identify a key task—Component-Level OBI Segmentation—based on a successful deciphering case from 2018. This task aims to help experts quickly identify specific components within OBIs, thereby accelerating the deciphering process. Accordingly, we propose a new model to accomplish this task. Our model leverages a small amount of annotated data and a large amount of weakly annotated data and incorporates expert-provided prior knowledge, i.e., stroke rules, to automatically segment OBI components. Additionally, we train a series of auxiliary classifiers to evaluate the segmentation results during the test stage. We also invite experts to conduct a professional assessment of the results, which we cross-validated against our proposed evaluation metrics. Experimental results demonstrate that our method can accurately and clearly present the segmented components to experts.

Code — https://github.com/hutt94/Component-Level-OBI-Segmentation

Introduction

The study of ancient scripts plays a fundamental role in archaeology and history (Keightley 1997; Flad 2008), yet understanding them poses significant challenges due to their antiquity. An exemplary case is Oracle Bone Inscriptions (OBIs), an ancient Chinese pictographic script dating back 3000 years, which serve as a valuable source for studying ancient Chinese civilizations, particularly the Shang civilization (Han 2012; Cheung 2018). However, despite extensive excavations yielding a large number of instinct OBIs (Cheung 2018), only approximately one-third have been fully deciphered by paleographers, leaving the remainder shrouded in mystery, preventing us further understanding the Shang civilization (Ochiai 2008).

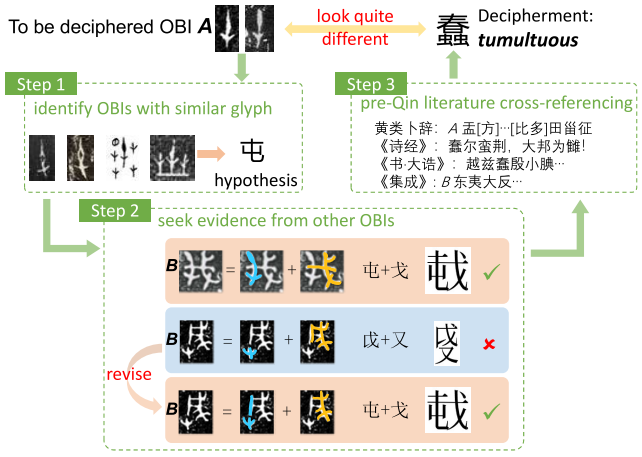

Figure 1: Deciphering process of OBI *tumultuous* proposed by Jiang (2018).

With remarkable achievements of artificial intelligence (AI) across various domains, such as AlphaGo (Silver et al. 2016), AlphaFold (Jumper et al. 2021), and Ithaca (Assael et al. 2022), an increasing number of researchers are now leveraging AI technologies to study OBIs (Wang et al. 2024a; Qiao et al. 2024; Wang et al. 2024b). These efforts broadly fall into three main directions: OBI rubbing denoising (Shi et al. 2022a,b; Jiang et al. 2023), OBI optical character recognition (OCR) (Fujikawa et al. 2023; Diao et al. 2023a,b), and OBI classification (Fu et al. 2022; Huang et al. 2019). Although these studies have, to some extent, accelerated OBI research, most of them are focused on the already deciphered OBIs and are far from achieving the ultimate goal of understanding and deciphering unknown OBIs.

Some approaches (Chang et al. 2022; Guan et al. 2024b,a) have innovatively attempted to decipher unknown OBIs by leveraging the structural relationships between modern Chinese characters and OBI glyph. For instance, Chang et al. (2022) proposed a cascade generative adversarial networks framework that incorporates bronze script and small seal script to simulate the evolutionary path of Chinese characters, ultimately predicting OBIs. Similarly, Guan et al.

*Corresponding author.

(2024b) employed a diffusion model to replicate the character evolution process. Such a strategy of deciphering OBIs through glyph evolution is an intuitive approach. Regretfully, as Prof. Huang Tianshu pointed out, these OBIs that exhibit direct correspondences with modern Chinese characters were basically deciphered by the late 20th century (Huang 2019). The remaining undeciphered OBIs are exceedingly difficult to decode solely through glyph analysis.

To illustrate this point, we briefly introduce a case where Jiang (2018) successfully deciphered the OBI for the term *tumultuous*. As shown in Figure 1, we denote the OBI to be deciphered as *A*. The deciphering process roughly includes three steps. **Step 1** involves identifying OBIs with similar glyph to *A* to make an initial semantic hypothesis. **Step 2** involves seeking evidence from other OBIs, e.g., *B* in Figure 1, to refine and support this hypothesis. **Step 3** involves cross-referencing pre-Qin literature to find corresponding corpus that can further validate the hypothesis from the **Step 2**, ultimately leading to the determination of *A*'s actual meaning. As the results show, the final deciphered meaning *tumultuous* differs significantly from *A* in terms of glyph structure, indicating that accurate interpretation cannot be achieved through glyph evolution alone.

In the aforementioned deciphering process, the most critical stage is **Step 2**. One of the key contributions of Jiang (2018) lies in revising the erroneous component segmentation of OBI *B* from previous study (Xie 2011) during this step, thereby enabling the textual evidence in **Step 3** to perfectly align with the corrected segmentation. As shown in Figure 1, incorrectly segmenting character *B* into different components can lead to a misunderstanding of its meaning, thereby affecting the interpretation of OBI *A*. Therefore, accurately segmenting OBIs to extract the target components is crucial. We termed this task **component-level OBI segmentation**. However, in practice, component-level OBI segmentation requires extensive expertise and is time-consuming due to the large volume of data. To address this, in this paper, we propose a new model to assist experts in this task, thereby accelerating the process of deciphering OBIs.

Component-level OBI segmentation is more challenging than traditional image segmentation (Zheng et al. 2021; Zhang et al. 2022) due to the following reasons: 1) the data annotation is limited because it relies on expert knowledge, 2) the components to be segmented have a high similarity to background strokes, making them harder to distinguish, and 3) without expert knowledge, it is difficult to evaluate the segmentation results. To address these challenges, this paper makes the following contributions:

- We propose a model that leverages both annotated and weakly annotated data, where the latter only indicates the category of the target component without specifically marking the component strokes within the OBI data. By utilizing weakly annotated data, our model can effectively segment various forms of components across different OBIs.
- Based on expert recommendations, we incorporate component stroke rules during training to constrain the segmentation results. Specifically, these stroke rules ensure the segmented results to adhere to the patterns identified by OBI experts, thereby improving the distinction between the components and the background strokes.
- During the training stage, we also train several auxiliary modules. These modules not only help constrain the segmentation results during the training stage, but also provide quantitative evaluation of the segmentation results during the test stage. We further invited experts to score the segmented results and compared these scores with the quantitative evaluations, validating the effectiveness of our assessments.

Background

OBI Tasks

In traditional studies of OBI (Jiang 2018; Chen 2019; Zhou 2019), research heavily relies on a small number of OBI experts who manually search through a vast amount of physical materials to decipher OBIs. This process is extremely time-consuming and labor-intensive. In recent years, an increasing number of researchers have turned to deep learning (DL) models to expedite the study of OBIs. These studies primarily focus on three tasks. The first task is OBI rubbing denoising (Shi et al. 2022a,b; Jiang et al. 2023). Most existing OBI data are in the form of rubbings. They often contain significant noise that hinders readability due to cracks on turtle shells and immature printing techniques. Some scholars have designed various networks to remove these noise. For example, Jiang et al. (2023) proposed OraclePoints where OBI is represented as hybrid neural representation to remove noise. The second task is OBI OCR (Fujikawa et al. 2023; Diao et al. 2023a,b), which involves extracting text from OBI images. Fujikawa et al. (2023) evaluated the recognition accuracy of different deep learning backbones for OBI OCR and developed a series of interfaces for OBI OCR. Diao et al. (2023a) explored zero-shot OBI OCR problems. The third task is OBI recognition (Fu et al. 2022; Huang et al. 2019), where DL models are used for character-level classification of OBIs. To facilitate this task, several datasets have been proposed (Huang et al. 2019; Fu et al. 2022).

Besides these works, some scholars have explored alternative approaches to assist in the deciphering of OBIs. For example, Hu et al. (2024) first introduced the component-level OBI retrieval task and collected a dataset OBI Component 20. They attempted to connect different OBIs through their components, aiding experts in identifying commonalities. Chang et al. (2022) and Guan et al. (2024b) attempted to directly predict undeciphered OBIs based on their glyph structures. Meanwhile, Qiao et al. (2024) recognized the pictographic nature of OBIs and explored a more intuitive method by representing these inscriptions as images to better convey their meanings.

Dataset

OBI Component 20 dataset is collected by Hu et al. (2024) for component-level OBI retrieval task. It consists of 9,245 OBI characters categorized into 20 classes based on their contained components (such as "foot", "child", "grass",

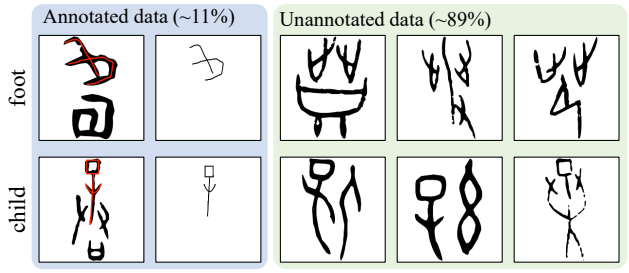

Figure 2: Some samples in OBI Component 20 dataset.

"water"). In each class, different forms of the same component are collected. Besides, 1,012 OBI characters have their components explicitly annotated, and these annotations are included as new OBI component data in the dataset, forming paired relationships with their corresponding OBI characters. For unannotated OBI characters, although the specific locations of the components is not annotated, the categories of the components they contain are known. Figure 2 displays some annotated and unannotated samples of components "foot" and "child".

Our annotation: To better achieve the component-level OBI segmentation, we extended the existing annotations of OBI Component 20 by incorporating stroke-based rules. Specifically, we enlisted experts to annotate the OBI component data across three dimensions: *Connected Domain Count (CDC)*, *Circle Count (CC)*, and *Intersection Count (IC)*. These three criteria are commonly used to assess the similarity between components. Specifically, *connected domain count* refers to the number of independent strokes within a component. *Circle count* indicates the number of closed circles formed by strokes within a component. *Intersection count* is the number of intersections between different strokes within a component. Figure 3 illustrates some sample annotations based on these dimensions. Based on these annotations and their distributions, we categorize the CDC into three classes (1, 2, and greater than 2), CC into three classes (0, 1, and greater than 1), and IC into five classes (0, 1, 2, 3, and greater than 3). Figure 4 shows the distribution of data across different categories.

Proposed Method

Problem Formulation

We have a dataset $\mathcal{I}^{ch} = \{I_i^{ch}\}_{i=1}^{n}$ consisting of n OBI character images, among which m images have been precisely annotated with the positions of components, denoted as $\mathcal{I}^{a} = \{I_i^{ch}\}_{i=1}^{m}$, while the remaining unannotated images are denoted as $\mathcal{I}^{u} = \{I_i^{ch}\}_{i=m+1}^{n}$. Due to the difficulty of annotation, we have $m \ll n$. Additionally, the annotated components from $\mathcal{I}^{a}$ are separately recorded in images containing only the respective components, forming the OBI component image set $\mathcal{I}^{co} = \{I_i^{co}\}_{i=1}^{m}$. Based on the components, $\mathcal{I}^{co}$ is categorized into c classes, and assigned corresponding labels $\{y_i^{co}\}_{i=1}^{m}$. Similarly, based on whether an OBI character contains a specific component, $\mathcal{I}^{ch}$ is also categorized into c classes, with labels $\{y_i^{ch}\}_{i=1}^{n}$. Furthermore,

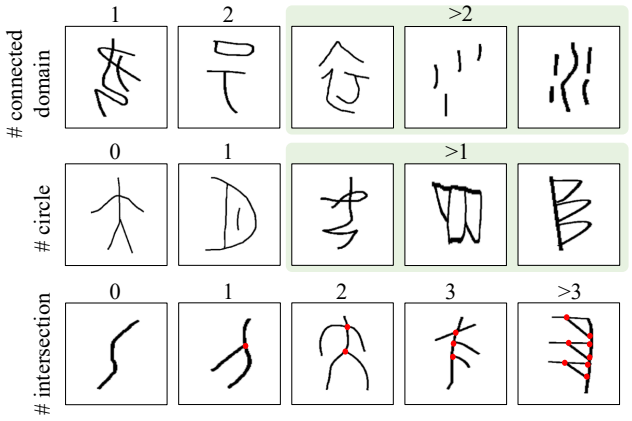

Figure 3: Some samples of our new annotations.

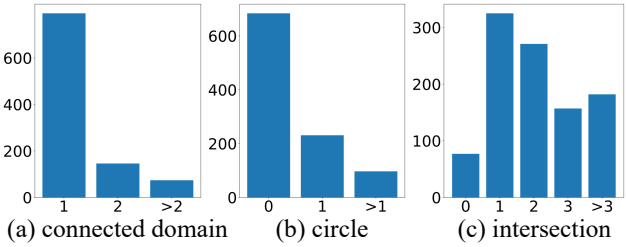

Figure 4: Statistics of stroke rule annotations.

for each component in $\mathcal{I}^{co}$, we have annotated the number of connected domains (labeled as $\{y_i^1\}_{i=1}^{m}$), the number of circles (labeled as $\{y_i^2\}_{i=1}^{m}$), and the number of intersections (labeled as $\{y_i^3\}_{i=1}^{m}$).

During the testing stage, given an OBI character image, the objective is to generate a blank image of the same size, where pixels corresponding to the positions of components are marked in red.

Overall Scheme

The proposed model consists of three main components: two encoders that respectively process OBI characters and OBI components, and one decoder that takes OBI character features to predict the segmentation results. For annotated data, we utilize $\mathcal{I}^{co}$ and their corresponding labels to supervise the learning of segmentation results. For unannotated data, we strengthen the association between annotated and unannotated data by constructing character-component-component triplets. Additionally, we introduce stroke rules to supervise the feature learning of components. The overall framework is illustrated in Figure 5.

Feature Extraction

Considering that OBI images contain limited information mainly focused on stroke features, we opt to use the powerful transformer-based model to extract relevant features. Specifically, we choose the Pyramid Vision Transformer (PVT) (Wang et al. 2021) as the encoder and decoder to ex-

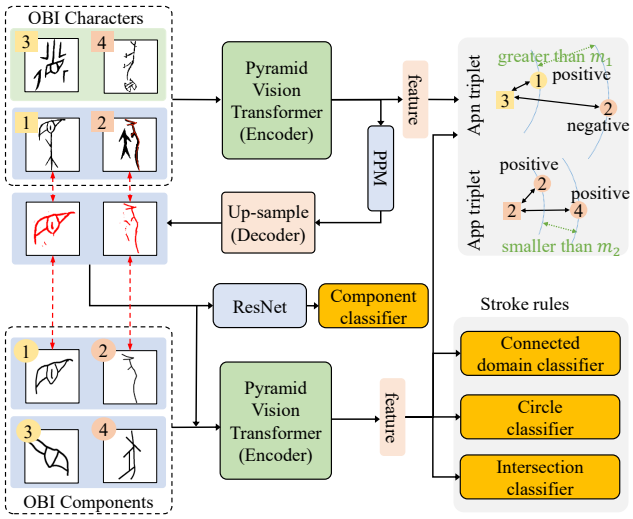

Figure 5: The proposed model consists of three main components: two encoders that respectively process OBI characters and OBI components, and one decoder that takes OBI character features to predict the segmentation results.

tract features from OBI characters and components, represented by f^{ch} and f^{co}, as follow:

$$f_i^{ch} = \text{Encoder}_{ch}(I_i^{ch}),\ f_i^{co} = \text{Encoder}_{co}(I_i^{co}). \quad (1)$$

Subsequently, we use the extracted features f^{ch} to reconstruct the segmentation result I^{pred} as follow:

$$I_i^{pred} = \text{Decoder}_{ch}(f_i^{ch}). \quad (2)$$

Annotated Data

For annotated data $\mathcal{I}^a$, we have two types of information, $\mathcal{I}^{co}$ and labels $\{y_i^{co}\}_{i=1}^m$, which serve as supervision signals. $\mathcal{I}^{co}$ can be used as the ground-truth segmentation result to supervise I^{pred}. On one hand, we can treat the prediction for each pixel in I^{pred} as a binary classification problem and define the following loss

$$\mathcal{L}_{rec} = \frac{1}{m}\sum_{i=1}^{m}\sum_{j} g_{ij}\log(p_{ij}), \quad (3)$$

where p_{ij} is the j-th pixel in I_i^{pred}, and $g_{ij} = 1$ indicates the corresponding pixel in I_i^{co} contains a component, and $g_{ij} = 0$ otherwise. On the other hand, we aim for the predicted pixels in I^{pred} to align as closely as possible with $\mathcal{I}^{co}$. Therefore, we define the loss

$$\mathcal{L}_{dice} = \frac{1}{m}\sum_{i=1}^{m}(1 - \frac{2\sum_j p_{ij}g_{ij}}{\sum_j p_{ij}^2 + \sum_j g_{ij}^2}). \quad (4)$$

To make the reconstructed I^{pred} closer to a complete component, we introduce a classifier Cls_0 to predict labels for I^{pred} and define the classification loss

$$\mathcal{L}_{cls0} = \frac{1}{m}\sum_{i=1}^{m} y_i^{co}\log(\text{Cls}_0(I_i^{pred})). \quad (5)$$

The overall loss function for annotated data is defined as

$$\mathcal{L}_{anno} = \mathcal{L}_{rec} + \mathcal{L}_{dice} + \mathcal{L}_{cls0}. \quad (6)$$

Unannotated Data

Since unannotated data $\mathcal{I}^u$ lacks direct ground-truth for component localization, we use labels to establish the relationship between $\mathcal{I}^u$, $\mathcal{I}^a$, and $\mathcal{I}^{co}$ indirectly by constructing character-component-component triplets at the feature level. Specifically, we construct two types of triplets: anchor-positive-negative triplets and anchor-positive-positive triplets.

Anchor-positive-negative (Apn) triplets. Although $\mathcal{I}^u$ lacks specific component localization annotations, its labels are known. We can use these labels to establish a connection between $\mathcal{I}^u$ and $\mathcal{I}^{co}$, enabling the learned features from unannotated data to be closer to the features learned from corresponding components. This, in turn, helps achieve better segmentation results during decoding. To achieve this purpose, we use the character $I_i^{ch} \in \mathcal{I}^{ch}$ as an anchor and construct an apn triplet. Specifically, we randomly select a positive sample I_j^{co+} from $\mathcal{I}^{co}$ with the same label as the anchor and a negative sample I_k^{co-} with a different label from the anchor. This process results in a triplet $(I_i^{ch}, I_j^{co+}, I_k^{co-})$. We aim for the distance between the anchor and the positive sample to be closer than the distance between the anchor and the negative sample. Thus, we formulate the following triplet loss function:

$$\mathcal{L}_{apn} = \frac{1}{n}\sum_{i=1}^{n} max(0, m_1 - D(f_i^{ch}, f_k^{co-}) + D(f_i^{ch}, f_j^{co+})), \quad (7)$$

where $D(\cdot,\cdot)$ denotes the distance between two features and m_1 controls the distance between positive and negative characters. In Eq. (7), we select data from both $\mathcal{I}^u$ and $\mathcal{I}^a$ as anchors to construct triplets. This could learn not only the relationship between $\mathcal{I}^u$ and $\mathcal{I}^{co}$ but also to more closely associate $\mathcal{I}^u$ with $\mathcal{I}^a$ through $\mathcal{I}^{co}$.

Anchor-positive-positive (App) triplets. For the same component, since there exist different forms, we aim to learn stable features from these diverse forms as much as possible. Specifically, we prefer less differences in features between a character and its corresponding component with different forms. To achieve this, we use the character as an anchor and construct an app triplet. In this triplet setup, we randomly select two positive samples I_j^{co+}, I_k^{co+} from $\mathcal{I}^{co}$ with the same label as the anchor. This process results in a triplet $(I_i^{ch}, I_j^{co+}, I_k^{co+})$. We aim for the distance between the anchor and the two positive samples to be as close as possible. Thus, we formulate the triplet loss function $\mathcal{L}_{app}$ as

$$\frac{1}{n}\sum_{i=1}^{n} max(0, \left|D(f_i^{ch}, f_j^{co+}) - D(f_i^{ch}, f_k^{co+})\right| - m_2), \quad (8)$$

where $|\cdot|$ represents absolute value and m_2 controls the distance between two positive components and anchor character. Similar to Eq. (7), Eq. (8) also utilizes data from both $\mathcal{I}^u$ and $\mathcal{I}^a$ as anchors to construct triplets, further strengthening the relationship between $\mathcal{I}^u$, $\mathcal{I}^a$, and $\mathcal{I}^{co}$.

Finally, the overall loss function for unannotated data is defined as the combination of these two loss functions

$$\mathcal{L}_{unan} = \mathcal{L}_{apn} + \mathcal{L}_{app}. \quad (9)$$

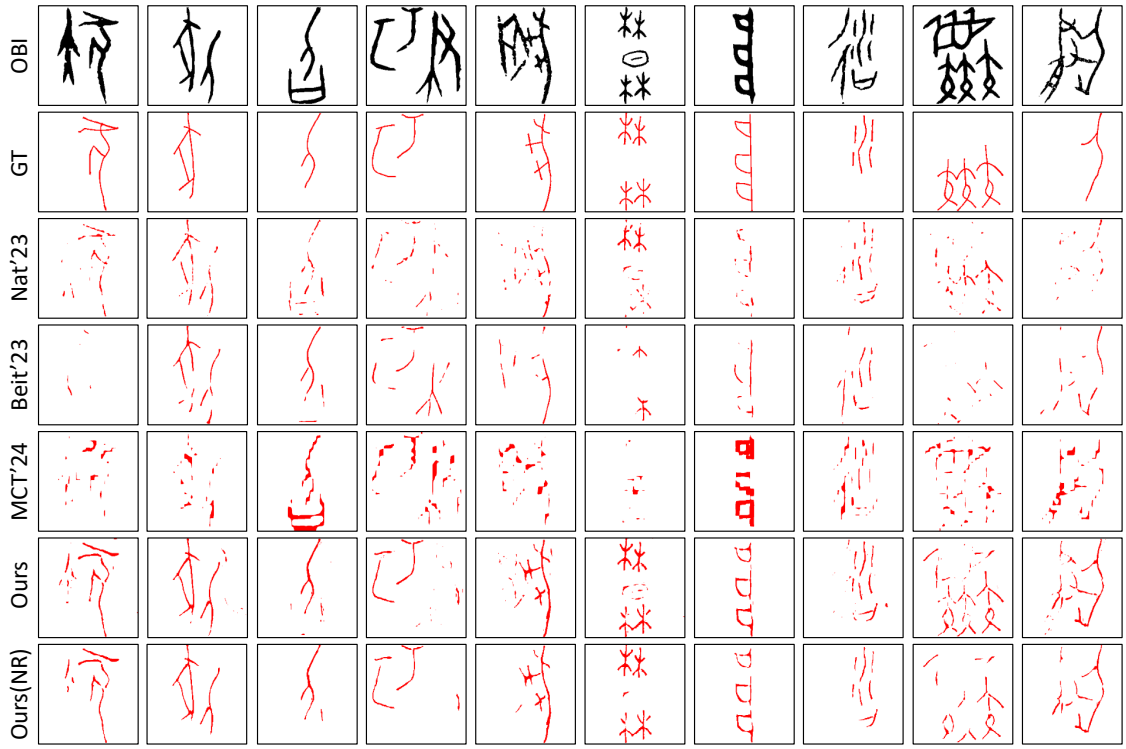

Figure 6: Segmentation results of all compared methods on the annotated data test set, where the segmented components are highlighted in red. "GT" represents the ground truth annotations.

Component Stroke Rules

Based on expert experience in identifying OBI components, although the same component may have different forms, they generally adhere to similar stroke rules, such as the number of connected domains, circles, and intersections. To ensure that the features of components capture these stroke rules, we introduce three classifiers Cls_1, Cls_2, and Cls_3 to impose constraints on the component features from the perspectives of connected domains, circles, and intersections. Additionally, we aim for the segmentation results I^{pred}, which is a also component, to also learn these rules. To achieve this, we use a component encoder to learn the features $f^{pred} = \mathrm{Encoder}_{co}(I^{pred})$ and feed these features to the classifiers. As a result, we obtain the following three classification loss functions:

$$\mathcal{L}_{cls1} = \frac{1}{2m}\sum_{i=1}^{m} y_i^1(\log(\mathrm{Cls}_1(f_i^{co})) + \log(\mathrm{Cls}_1(f_i^{pred})),$$
$$\mathcal{L}_{cls2} = \frac{1}{2m}\sum_{i=1}^{m} y_i^2(\log(\mathrm{Cls}_2(f_i^{co})) + \log(\mathrm{Cls}_2(f_i^{pred})),$$
$$\mathcal{L}_{cls3} = \frac{1}{2m}\sum_{i=1}^{m} y_i^3(\log(\mathrm{Cls}_3(f_i^{co})) + \log(\mathrm{Cls}_3(f_i^{pred})). \tag{10}$$

Finally, the overall loss function for stroke rule preserving is defined as the combination of these three loss functions:

$$\mathcal{L}_{stroke} = \mathcal{L}_{cls1} + \mathcal{L}_{cls2} + \mathcal{L}_{cls3}. \tag{11}$$

Combining these three parts, the overall loss function is

$$\mathcal{L} = \mathcal{L}_{anno} + \alpha\mathcal{L}_{unan} + \beta\mathcal{L}_{stroke}, \tag{12}$$

where α and β are hyper-parameters to balance three items.

Experiment

Experimental Settings

Data partition. The OBI Component 20 dataset consists of 1,012 annotated characters and 8,233 unannotated characters. For training our model, we used 75% of the annotated data as the training set, with the remaining 25% reserved for test. Additionally, all unannotated data was included in the model training stage.

Compared methods. We select two supervised methods Beit (Wang et al. 2023) and Nat (Hassani et al. 2023), and a weakly supervised method MCT (Xu et al. 2024) as compared methods. For Beit and Nat, we initialize their parameters using pre-trained models publicly provided by authors and fine-tune them using 75% of the annotated OBI data. For MCT, the training setting is same as our model.

Evaluation metrics. Given the unique nature of the component-level OBI segmentation task, we evaluated the segmentation results from four perspectives: (1) Standard metrics used in image segmentation, including mean Intersection over Union (mIoU), precision, and recall are employed. (2) We visualize the segmentation results to facilitate intuitive assessment. (3) We train additional classifiers

Metrics		Nat'23	Beit'23	MCT'24	Ours	Ours (NR)	Baseline
mIoU (%)		58.76	56.71	53.01	<u>61.45</u>	**61.64**	-
precision (%)		<u>67.71</u>	64.23	56.25	66.11	**67.94**	-
recall (%)		64.36	61.95	62.77	**75.50**	<u>72.05</u>	-
Accuracy (%)	Component	21.46	23.18	12.45	<u>33.91</u>	**35.19**	81.11
	CDC	30.90	27.04	24.46	**40.34**	<u>37.76</u>	69.96
	CC	59.23	62.23	66.52	<u>66.95</u>	**67.38**	83.26
	IC	56.22	60.94	**73.39**	<u>62.23</u>	60.08	84.98

Table 1: Quantitative metrics for component segmentation on the test set of all compared methods.

for components, CDC (connected domain count), CC (circle count), and IC (intersection count), using their accuracy to further evaluate the segmentation results. (4) Experts are invited to conduct a professional evaluation of the segmentation results based on three criteria: recognition, completeness, and noise level. We cross-validate these four evaluation methods to derive quantitative metrics more suited to this specific task.

Implementation details. We use an NVIDIA RTX 3090 to implement our method with the popular PyTorch toolkit. For our backbone, all PVT modules are PVT medium pretrained on ImageNet. Our model is trained using Adam-W optimizer with weight decay of 4e-2, batch size of 32, for 100 epochs with 5 warm up epochs. Using cosine scheduler to adjust learning rate from 3e-3 to 4e-1. α and β are set to 1. Two threshold m_1 and m_2 are set to 1.

Standard Metrics

Table 1 presents the mIoU, precision, and recall metrics for all methods on the test set. Our approach significantly outperforms the comparison methods in terms of mIoU and recall. However, in terms of precision, there is little difference among the methods, except for MCT. Nonetheless, we believe that due to the unique annotation style of OBI Component 20, these three metrics are not entirely suitable for evaluating the segmentation results. Specifically, the strokes in OBI Component 20 are relatively thick, while the annotated components have finer strokes. This discrepancy can result in situations where the segmentation is correct, but does not overlap with the annotated components, meaning the segmentation selects different pixels within the same stroke compared to the annotations. We will discuss this issue in the next section.

Visual Results Analysis

To provide a more intuitive comparison of all methods, we present the segmentation results in Figure 6. It can be observed that although Beit, Nat, and MCT did not perform poorly according to the quantitative metrics in the previous section, their segmentation results are difficult to be identified as correct components. Specifically, these methods tend to fragment the strokes into small segments, rendering the components challenging to be recognized. In contrast, our proposed method successfully segments more complete components. We further applied noise removal, denoted as Ours (NR), eliminating strokes that occupy less than 0.07% of the image area. As shown, Ours (NR) is able to present the segmented components more clearly.

Additionally, our method sometimes produces slightly thicker strokes in the segmented components, which may lead to discrepancies with the groundtruth and negatively impact metrics like mIoU. However, this variation in stroke thickness does not actually hinder component recognition. Therefore, the three metrics discussed in the previous section are not well-suited for evaluating OBI segmentation task presented in this paper.

Classification Accuracy

We further evaluate the segmentation results using the four classifiers trained during the training stage. The results are presented in Table 1. The baseline represents the classification accuracy of these four classifiers on the groundtruth in the test set, which is relatively high, indicating a certain degree of reliability. It can be observed that the results from the component classifier are more consistent with the visual results. However, the other three classifiers did not perform well in distinguishing between all methods. This may suggest that while these classifiers can constrain features during the training stage, they may not be suitable for evaluating the results in the test stage.

Expert Evaluation

To more accurately assess the segmentation results, we invited experts to score the segmented components on three levels: Recognition (R), Completeness (C), and Noise (N), with scores ranging from 1 to 5. Recognition indicates how easily the segmentation result can be recognized, with higher scores being better. Completeness reflects the integrity of the segmentation, with higher scores indicating better completeness. Noise represents the amount of noise in the segmentation, with lower scores being preferable. The results are presented in Table 2. It can be seen that our proposed method significantly outperforms the comparison methods in terms of recognition rate. This is consistent with the visual results and the component classification results, confirming reliability of the later as a quantitative evaluation metric. Furthermore, with additional noise removal, the recognition rate of our method can be further improved, although the completeness of the components may decrease. This also indicates that, even with less complete components, experts are still able to recognize them effectively when noise is minimized.

Criteria	Nat	Beit	MCT	Ours	Ours (NR)
R ↑	1.81	1.81	1.48	3.31	**3.41**
C ↑	1.78	1.97	1.56	**3.70**	3.59
N ↓	3.44	2.76	4.16	2.55	**1.75**

Table 2: Expert evaluation.

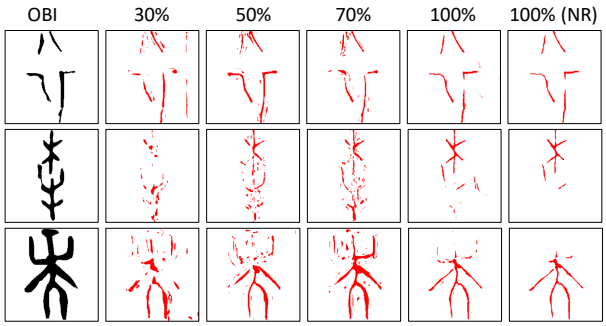

Figure 7: Segmentation results with varying amounts of annotated data.

Metrics	full	w./o. $\mathcal{L}_{unan}$	w./o. $\mathcal{L}_{stroke}$
Component Acc.	**33.91**	33.04	24.03
CDC Acc.	40.34	**41.2**	33.05
CC Acc.	66.95	**69.52**	66.52
IC Acc.	62.23	**63.94**	62.66

Table 3: Ablation study of the proposed method.

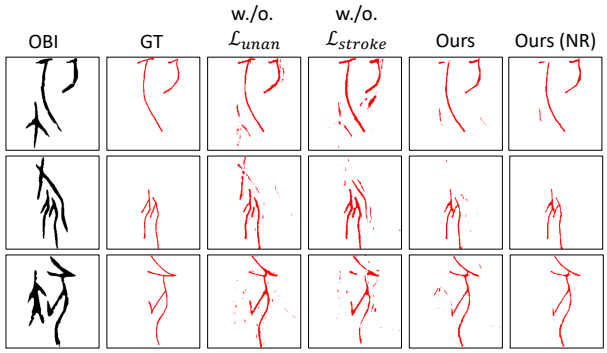

Figure 8: Segmentation results of different variants.

Proportion of Annotated Data

To validate the impact of the proportion of annotated data on segmentation performance, we reduce the ratio of annotated data while keeping the unannotated data unchanged. The results are illustrated in Figure 7. It can be observed that as the proportion of annotated data decreases, the noise in the segmentation results increases. With 70% of the annotated data, the results still exhibit recognizable and complete components; at 50%, noise increases significantly, and the strokes of the components begin to exhibit omissions; with only 30%, the components become nearly unrecognizable.

Ablation Study

To validate the effectiveness of introducing unannotated data and stroke rules, we conducted experiments by removing $\mathcal{L}_{unan}$ and $\mathcal{L}_{stroke}$ from our model. The quantitative results are shown in Table 3. It can be observed that the proposed full model achieves the highest component classification accuracy. We visualized the segmentation results of these variants in Figure 8. As shown, if unannotated data is excluded from training, the segmentation results tend to include more noise. In contrast, the introduction of stroke rules effectively prevents the appearance of unnecessary large strokes in the segmentation results. This demonstrates the effectiveness of these two loss functions.

Conclusion and Discussion

In this paper, we have introduced the component-level OBI segmentation task, motivated by the challenges encountered in the process of deciphering OBIs. To achieve this task, we have proposed a weakly supervised approach that leverages a larger amount of unannotated data while incorporating OBI stroke rules during the model training stage. This approach has addressed the issues of limited annotated data and the high difficulty of segmentation in this task. Additionally, we have designed various metrics to evaluate the quality of the segmentation results and have conducted cross-validation among these metrics. Experimental results have demonstrated that by incorporating stroke rules and utilizing unannotated data, our method has achieved segmentation outcomes that meet expert expectations.

Limitation and future work: In this paper, the segmentation task assumes that each OBI contains only one target component. However, in reality, many OBI characters consist of multiple components. For example, the second OBI in Figure 6 contains both the "dog" and "knife" components, but Hu et al. (2024) classified it only under the"dog" category. While our segmentation results have identified both components, they are not able to distinguish between them. In future work, we plan to develop new methods to further differentiate between the different OBI components within a single OBI character, without requiring additional annotations. This will allow for a more intuitive presentation of the segmentation results to experts, thereby accelerating the deciphering of OBIs.

Broader social impact: The research on components within OBI characters presented in this paper can be broadly applied to other pictographic scripts, such as the cuneiform writing of the Sumerian civilization and the Egyptian hieroglyphs of ancient Egypt. Many of these scripts also suffer from a scarcity of annotations. The model proposed in this paper can serve as a reference for scholars studying these scripts, contributing significantly to advancing research on ancient scripts globally.

Acknowledgements

This work was supported in part by the NSFC / Research Grants Council (RGC) Joint Research Scheme under grant:

N_HKBU214/21, the General Research Fund of RGC under the grants: 12202622, 12201323, and the RGC Senior Research Fellow Scheme under the grant: SRFS2324-2S02. We express our gratitude to all the anonymous reviewers for their valuable feedback. Special thanks are extended to Mr. Li Chang-Xing for his assistance with the experiments.

References

Assael, Y.; Sommerschield, T.; Shillingford, B.; Bordbar, M.; Pavlopoulos, J.; Chatzipanagiotou, M.; Androutsopoulos, I.; Prag, J.; and de Freitas, N. 2022. Restoring and attributing ancient texts using deep neural networks. *Nature*, 603(7900): 280–283.

Chang, X.; Chao, F.; Shang, C.; and Shen, Q. 2022. Sundialgan: A cascade generative adversarial networks framework for deciphering oracle bone inscriptions. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1195–1203.

Chen, J. 2019. Variant Forms of the Character 'Chè' in Oracle Bone and Bronze Inscriptions: Another Example of Character Interpretation Based on Differences in the Categories of Divinatory Inscriptions. http://www.fdgwz.org.cn/Web/Show/4382. Accessed: 2019-02-05.

Cheung, C. 2018. Chinese Oracle Bones - The Chinese History That Is Written in Bone. https://www.sapiens.org/archaeology/chinese-oracle-bones-history/. Accessed: 2018-01-23.

Diao, X.; Shi, D.; Li, J.; Shi, L.; Yue, M.; Qi, R.; Li, C.; and Xu, H. 2023a. Toward Zero-shot Character Recognition: A Gold Standard Dataset with Radical-level Annotations. In *Proceedings of the 31st ACM International Conference on Multimedia*, 6869–6877.

Diao, X.; Shi, D.; Tang, H.; Qiang, S.; Li, Y.; Wu, L.; and Xu, H. 2023b. RZCR: zero-shot character recognition via radical-based reasoning. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, 654–662.

Flad, R. K. 2008. Divination and power: a multiregional view of the development of oracle bone divination in early China. *Current Anthropology*, 49(3): 403–437.

Fu, X.; Yang, Z.; Zeng, Z.; Zhang, Y.; and Zhou, Q. 2022. Improvement of Oracle Bone Inscription Recognition Accuracy: A Deep Learning Perspective. *ISPRS International Journal of Geo-Information*, 11(1): 45.

Fujikawa, Y.; Li, H.; Yue, X.; Aravinda, C.; Prabhu, G. A.; and Meng, L. 2023. Recognition of oracle bone inscriptions by using two deep learning models. *International Journal of Digital Humanities*, 5(2): 65–79.

Guan, H.; Wan, J.; Liu, Y.; Wang, P.; Zhang, K.; Kuang, Z.; Wang, X.; Bai, X.; and Jin, L. 2024a. An open dataset for the evolution of oracle bone characters: EVOBC. *arXiv preprint arXiv:2401.12467*.

Guan, H.; Yang, H.; Wang, X.; Han, S.; Liu, Y.; Jin, L.; Bai, X.; and Liu, Y. 2024b. Deciphering Oracle Bone Language with Diffusion Models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 15554–15567.

Han, J. 2012. *Chinese characters*. Cambridge University Press.

Hassani, A.; Walton, S.; Li, J.; Li, S.; and Shi, H. 2023. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6185–6194.

Hu, Z.; Cheung, Y.-m.; Zhang, Y.; Zhang, P.; and Tang, P.-l. 2024. Component-Level Oracle Bone Inscription Retrieval. In *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 647–656.

Huang, S.; Wang, H.; Liu, Y.; Shi, X.; and Jin, L. 2019. OBC306: A large-scale oracle bone character recognition dataset. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 681–688. IEEE.

Huang, T. 2019. Huang, Tianshu: Oracle Bone Script Begins with Literacy. https://www.lifeweek.com.cn/article/79657. Accessed: 2019-10-07.

Jiang, R.; Liu, Y.; Zhang, B.; Chen, X.; Li, D.; and Han, Y. 2023. OraclePoints: A Hybrid Neural Representation for Oracle Character. In *Proceedings of the 31st ACM International Conference on Multimedia*, 7901–7911.

Jiang, Y. 2018. Interpreting the 'Tumultuous' in Oracle Bone and Bronze Inscriptions: A Comprehensive Analysis of Relevant Issues. http://www.fdgwz.org.cn/Web/Show/4472. Accessed: 2019-10-23.

Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. 2021. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873): 583–589.

Keightley, D. N. 1997. *Graphs, words, and meanings: Three reference works for Shang oracle-bone studies, with an excursus on the religious role of the day or sun*. JSTOR.

Ochiai, A. 2008. Reading History from Oracular Bone Inscriptions. *Chikuma Shobo*.

Qiao, R.; Yang, L.; Pang, K.; and Zhang, H. 2024. Making Visual Sense of Oracle Bones for You and Me. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12656–12665.

Shi, D.; Diao, X.; Shi, L.; Tang, H.; Chi, Y.; Li, C.; and Xu, H. 2022a. CharFormer: A glyph fusion based attentive framework for high-precision character image denoising. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1147–1155.

Shi, D.; Diao, X.; Tang, H.; Li, X.; Xing, H.; and Xu, H. 2022b. RCRN: Real-world Character Image Restoration Network via Skeleton Extraction. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1177–1185.

Silver, D.; Huang, A.; Maddison, C. J.; Guez, A.; Sifre, L.; Van Den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. 2016. Mastering the game of Go with deep neural networks and tree search. *nature*, 529(7587): 484–489.

Wang, P.; Zhang, K.; Liu, Y.; Wan, J.; Guan, H.; Kuang, Z.; Wang, X.; Jin, L.; and Bai, X. 2024a. An open dataset for oracle bone script recognition and decipherment. *arXiv preprint arXiv:2401.15365*.

Wang, P.; Zhang, K.; Wang, X.; Han, S.; Liu, Y.; Jin, L.; Bai, X.; and Liu, Y. 2024b. Puzzle Pieces Picker: Deciphering Ancient Chinese Characters with Radical Reconstruction. In *International Conference on Document Analysis and Recognition*, 169–187. Springer.

Wang, W.; Bao, H.; Dong, L.; Bjorck, J.; Peng, Z.; Liu, Q.; Aggarwal, K.; Mohammed, O. K.; Singhal, S.; Som, S.; et al. 2023. Image as a foreign language: Beit pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19175–19186.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, 568–578.

Xie, M. 2011. An Analysis of the Character 'Yue' in Bronze Inscriptions. *Chinese Character*, 37: 135–145.

Xu, L.; Bennamoun, M.; Boussaid, F.; Laga, H.; Ouyang, W.; and Xu, D. 2024. MCTformer+: Multi-Class Token Transformer for Weakly Supervised Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12): 8380–8395.

Zhang, B.; Tian, Z.; Tang, Q.; Chu, X.; Wei, X.; Shen, C.; et al. 2022. Segvit: Semantic segmentation with plain vision transformers. *Advances in Neural Information Processing Systems*, 35: 4971–4982.

Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P. H.; et al. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6881–6890.

Zhou, Z. 2019. Deciphering Chinese Character Tai (a Type of Slave) in the Unearthed Documents. *Bulletin of the Institute of History and Philology Academia Sinica*, 90(3): 367–398.