FORM GENERATION OF LOST ANCIENT CHINESE CHARACTERS USING COMPONENT PRIORS BY DIFFUSION MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

The form generation of lost ancient Chinese characters is a meaningful but challenging task. The existing methods for form generation of lost ancient Chinese characters do not utilize domain-specific knowledge fully and provide inadequate guidance in the generation process. In this paper, we propose a diffusion-based method to improve the form generation of lost ancient Chinese characters by domain-specific knowledge of components and enhancing guidance in the generation process. In particular, to improve generation quality by component priors, we introduce a component pre-training stage to learn evolutionary patterns of components. To enhance guidance in the generation process, we inject conditional features into the bottleneck of the U-Net to improve the similarity between the generated and target character forms. In addition, to better generate the details of components, we employ a unified patch sampling strategy across stages, ensuring a direct correspondence between local component features. Extensive experiments on the Font-607 dataset show that our method improves the quality of ancient Chinese character generation. For example, our method reduces the LPIPS score from 0.2517 to 0.2346, and increases the cosine similarity from 0.7841 to 0.8094 in the Seal Script generation task. Code is available at `https://anonymous.4open.science/r/Anc-Diff-81EA`.

1 INTRODUCTION

Generating the form of lost ancient Chinese characters is a meaningful task. Ancient Chinese characters, such as Oracle Bone Inscription, constitute the primary records of early human civilization so they are of great importance for human civilization research Dai et al. (2007). However, the form of some ancient Chinese characters during a particular era is lost due to complex historical reasons. As illustrated in Figure 1, the form generation of ancient Chinese characters is to generate the lost form of characters in a particular era by the evolutionary patterns of this character across other eras. The success of generating lost forms of ancient Chinese characters has great potential for advancing research in human civilization.

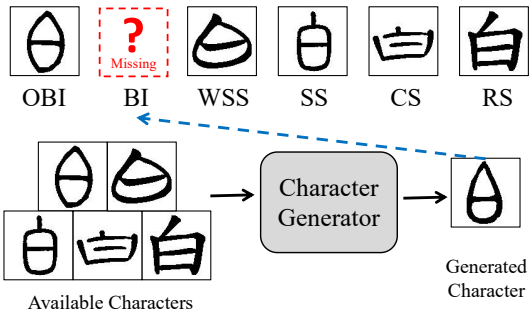

Figure 1: Illustration of the task. OBI, BI, WSS, SS, CS, and RS are the abbreviations of Oracle Bone Inscriptions, Bronze Inscriptions, Warring States script, Seal script, Clerical script, and Regular script, respectively.

The diffusion-based method is the most promising method for generating the lost form of ancient Chinese characters. The conventional method of few-shot font generation is well-suited for transferring styles between modern fonts with consistent structures Yang et al. (2024); Park et al. (2021b); Xie et al. (2021). However, these methods do not perform well enough when dealing with ancient Chinese characters because form variations of ancient Chinese characters is great. Early GAN-based methods suffer from training instability and model collapse, leading to distorted struc-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

tures and blurry details Mirza & Osindero (2014). For instance, Sundial-GAN Chang et al. (2022) is a cascaded generative adversarial network that uses different GAN methods at different stages. However, errors inevitably propagate and accumulate across successive stages. Recently, diffusion models have emerged as a powerful alternative. By leveraging a progressive denoising process, diffusion models exhibit enhanced training stability and a superior capacity for modeling complex structural variations.

However, the diffusion-based method does not perform as well as expected in form generation of lost ancient Chinese characters due to special properties of ancient Chinese characters. First, the general diffusion-based method treats the form generation of lost ancient Chinese characters as a general image-to-image translation task, overlooking the fact that Chinese characters are composed of distinct components. Second, the general-based diffusion method provides inadequate guidance in the generation process, resulting in that structural constraints are insufficient. Third, although the general diffusion-based method can capture the overall style, it does not pay enough attention to fine-grained local details which are critical in the form generation of lost ancient Chinese characters.

To address these challenges caused by the special properties of ancient Chinese characters, we propose a novel diffusion-based method, named Anc-Diff, to generate the lost form of ancient Chinese characters. In particular, to fully utilize domain-specific knowledge, our method incorporates a component pre-training stage, by which our method captures the structural priors. To enhance the guidance of the conditional images, our method introduces a decoupled guidance control module, by which our method injects conditional features into the U-Net Ronneberger et al. (2015) bottleneck so that guidance is enhanced. To better capture local features, our method employs a unified patch sampling strategy across two different stages, ensuring that our method maintains a consistent focus on specific component regions. In summary, our contributions are as follows:

- To improve the form generation quality of lost Chinese characters, we introduce a pre-training stage to capture priors of components. Our method utilizes these learned priors (fundamental stroke and radical patterns) to generate the form of lost ancient Chinese characters, resulting in the improvement of generation quality.
- To enhance guidance in the generation process, we propose a decoupled guidance control module. The module decouples the conditional feature extraction from the main encoding path and injects conditional features at the bottleneck of the U-Net. This design separates conditional feature extraction from noisy target encoding to improve the similarity between generated and target character forms.
- Extensive experiments are conducted on the character evolution dataset, Font-607. Experimental results demonstrate the effectiveness of our method in generating high-quality ancient characters. For example, our method reduces the LPIPS score by 0.0171, and increases the cosine similarity by 0.0253 in the Seal Script generation task.

2 Related Work

2.1 Character Generation

Early methods for character form generation primarily rely on image-to-image translation methods such as StarGAN Choi et al. (2018), Pix2Pix Isola et al. (2017) and CycleGAN Zhu et al. (2017). However, these methods often fail to preserve the topological integrity of Chinese characters, leading to distorted strokes. Sundial-GAN Chang et al. (2022) is a cascaded generative adversarial network, which connects multiple GANs in series. Subsequently, few-shot font generation emerges and can be divided into two categories: global feature methods and component feature methods. Global feature methods like EMD Zhang et al. (2018b) and AGISNet Gao et al. (2019) extract the global style and content features of characters to generate target character images. FontDiffuser Yang et al. (2024) and MSD-Font Fu et al. (2024) utilize the diffusion model to integrate the style and content features, achieving better results. However, these methods often struggle with fine-grained local details. Component feature methods such as DM-Font Cha et al. (2020), LF-Font Park et al. (2021b) and MX-Font Park et al. (2021a) decompose characters into multiple components and learn the localized features. Diff-Font He et al. (2024) takes the stroke information into account and uses a diffusion model for font generation. Although effective for modern fonts, component feature methods do not perform well in form generation of ancient Chinese characters due to the extreme

shape differences between forms. Recently, OBSD Guan et al. (2024) uses a conditional diffusion model with localized structural sampling to decipher the Oracle Bone Inscriptions. Our method introduces component pre-training and an improved patch sampling strategy for ancient character generation.

2.2 Diffusion Model

Recently, diffusion models have achieved rapid development in vision generation tasks, surpassing generative adversarial networks. DDPM Ho et al. (2020) first establishes the framework for high-quality image synthesis through a sequential denoising process. To improve efficiency, DDIM Song et al. (2021) introduces a non-Markovian sampling strategy, significantly accelerating inference while maintaining synthesis quality. To alleviate the computational burden of high-resolution generation, LDM Rombach et al. (2022) shifts the diffusion process from pixel space to a compressed latent space. To guide the diffusion process toward specific targets, several controllable diffusion models are proposed. ControlNet Zhang et al. (2023) adds a trainable branch to the diffusion backbone for precise control, while T2I-Adapter Mou et al. (2024) uses a more lightweight plug-and-play module to inject spatial hints. Furthermore, LayoutDiffusion Zheng et al. (2023) focuses on using semantic layouts to guide the position and arrangement of generated objects. In the field of image-to-image translation, diffusion-based methods like Palette Saharia et al. (2022) and BBDM Li et al. (2023) demonstrate superior performance in maintaining pixel-level details. However, these general-purpose models do not take the domain-specific knowledge of form generation of ancient Chinese characters, resulting in the performance of these methods can be improved further.

3 Method

3.1 Overview

The overall workflow of our method is illustrated in Figure 2, which is structured into three progressive stages: component pre-training stage, character training stage, and restoration stage.

In the component pre-training stage, our method pre-trains diffusion model on a specialized dataset containing component pairs (i.e., strokes and radicals). By decomposing component pairs into patch-based sequences, our method can capture domain-specific priors of components.

In the character training stage, our method extends the patch-based learning to the entire character. By inheriting the same patch sampling strategy, our method maintains consistency in local features.

In the character training stage, to enhance guidance in the generation process, our method introduces a decoupled guidance control module. By decoupling the feature extraction and injecting conditional features into the U-Net bottleneck, our method enhances the guidance in the generation process.

In the restoration stage, a sliding window is employed to generate the character via local patches. Specifically, the model generates each local patch independently and then aggregates them to form the complete character, which ensures that local component details are preserved while maintaining global structure.

Technically, our method implements the form generation task of ancient Chinese characters through the diffusion model. In the forward process of diffusion model, Gaussian noise is incrementally injected into the initial character image x_0 across T timesteps.

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\epsilon, \tag{1}$$

where $\bar{\alpha}_t := \prod_{s=1}^{t} \alpha_s$ and $\alpha_t := 1 - \beta_t$ ($\beta_1, \cdots, \beta_T$ is a pre-defined variance schedule) . $\epsilon \sim \mathcal{N}(0, I)$ is the noise.

Conversely, the reverse process of diffusion model aims to reconstruct the clean image by training a neural network $\boldsymbol{\epsilon}_\theta$ to estimate the added noise, thereby iteratively approximating the preceding state x_{t-1} from x_t. The loss can be expressed as:

$$\mathcal{L} = \mathbb{E}_{t, x_0, \boldsymbol{\epsilon}} \left[\left\| \boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\boldsymbol{\epsilon}, t) \right\|_2^2 \right]. \tag{2}$$

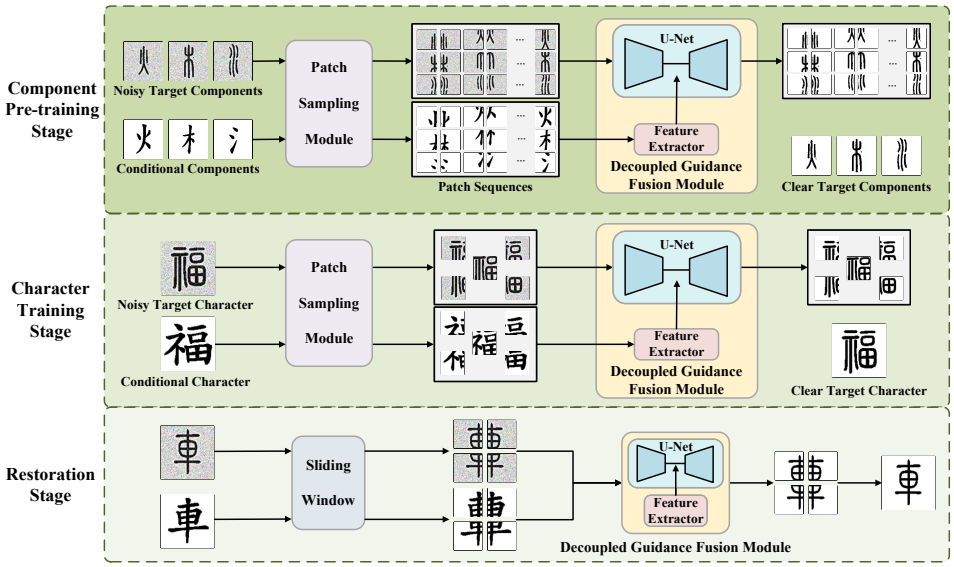

Figure 2: Overview of our proposed method, which consists of three stages.

3.2 Component Pre-training Stage

A component pre-training stage is introduced to learn components priors of ancient Chinese characters. Considering that ancient Chinese characters are composed of distinct components, components priors (evolutionary patterns of components) are crucial for generating the form of lost ancient Chinese characters. Taking the generation from Regular script to Seal script as an example, Figure 2 illustrates how our method learns components priors.

A diffusion model is utilized to learn the component priors by predicting and removing noise from a noisy input image to reconstruct a clear image. To this end, we first construct a component dataset consisting of paired Seal and Regular character components. The Seal character component serves as the target image, while the Regular character component serves as the conditional image.

Consider a pair consisting of a noisy Seal character component x and a clear Regular character component c. The patch sampling module is first applied to segment each component into a sequence of patches, denoted as $\{x^{(i)}\}_{i=1}^{n}$ and $\{c^{(i)}\}_{i=1}^{n}$ respectively. This processing enables our method to better capture fine-grained local features. Consequently, these sequences are fed into a decoupled guidance control module. Specifically, the sequence of noisy patches $\{x^{(i)}\}$ is fed directly into the U-Net, while the corresponding sequence of clear patches $\{c^{(i)}\}$ (serving as the condition) is processed through a feature extractor. The learned features are then injected into the U-Net at the bottleneck. Through this process, our method learns to turn noisy target components into clear ones.

The objective of this stage is to minimize the patch-level denoising loss, defined as follows:

$$\mathcal{L} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}_{t,{x_0}^{(i)},\boldsymbol{\epsilon}^{(i)},c^{(i)}} \left[\left\| \boldsymbol{\epsilon}^{(i)} - \boldsymbol{\epsilon}_\theta({x_t}^{(i)}, t, c^{(i)}) \right\|_2^2 \right], \tag{3}$$

where ${x_t}^{(i)} = \sqrt{\bar{\alpha}_t}{x_0}^{(i)} + \sqrt{1-\bar{\alpha}_t}\epsilon^{(i)}$.

By minimizing the loss function, the model acquires the ability to predict the noise within a noisy input image, thereby enabling the reconstruction of clear components from their noisy versions. Through this process, our method captures the component priors of ancient characters.

3.3 Character Training Stage

Building upon the component priors acquired in the component pre-training stage, the character training stage extends the learning process to the entire Seal character x and the corresponding Regular character character c. In this stage, our method is initialized with the weights from the pre-training stage to leverage the learned components priors.

To better capture the local features and ensure the integrity of the components, we use the same patch sampling module as in the previous stage. Then, for the Seal character x and the Regular character c, our method obtains the corresponding patch sequences $\{x^{(i)}\}_{i=1}^{n}$ and $\{c^{(i)}\}_{i=1}^{n}$. As shown in Figure 3, for each character image, four patches are sampled from the corners. The patch size is intentionally set to be slightly larger than half of the image dimensions to ensure full overlapping coverage of the entire character. Furthermore, additional patches are randomly cropped to enhance data diversity and ensure that all internal components are effectively captured.

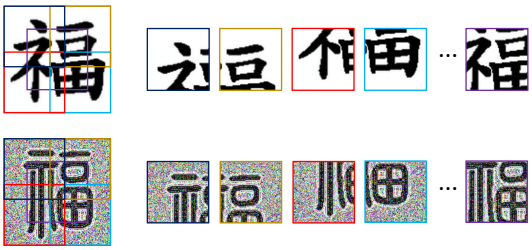

Figure 3: Convert the images into patch sequences.

To enhance the guidance of the generation process, the patch sequences are fed into the decoupled guidance control module. The sequence of noisy Seal character patches $\{x^{(i)}\}$ is fed directly into the U-Net. Meanwhile, the corresponding Regular character patches are first processed by the feature extractor $f(\cdot)$ to capture conditional features, which are then injected into the U-Net bottleneck. Unlike conventional methods that typically concatenate conditional features with target features at the input layer, our method injects conditional features at the bottleneck Guan et al. (2024); Dhariwal & Nichol (2021). Early fusion forces the encoder to process noisy and clean data simultaneously, which causes feature interference. In contrast, injecting features at the bottleneck avoids interference during feature extraction. The high-level features at this layer provide rich semantic information and focus on the overall structure of the character. This provides more powerful and reliable guidance for the generation process.

The training objective in this stage remains consistent with Eq. (3), but is now applied to the entire character images. Through this process, our method refines the global character structure while effectively preserving the fine-grained component priors learned in the component pre-training stage.

3.4 Restoration Stage

In the restoration stage, the objective is to generate the target form of ancient character given the conditional input images. Here, we still take the generation from the Regular character to the Seal character as an example. Unlike previous stages which perform single-step noise prediction, the restoration stage employs an iterative denoising process. Starting from Gaussian noise, our method progressively removes predicted noise guided by the conditional Regular characters features.

In this stage, a sliding window is utilized to segment the target Seal character image x and the conditional Regular character image c into sequences of patches $\{x^{(i)}\}_{i=1}^{N}$ and $\{c^{(i)}\}_{i=1}^{N}$, where N denotes the total number of patches per image. These regularly sampled patches are processed in pairs $(x^{(i)}, c^{(i)})$ through an iterative denoising process. Specifically, at each timestep t, the model predicts the noise $\epsilon_\theta(x_t^{(i)}, t, c^{(i)})$ within the current noisy patch $x_t^{(i)}$. Then, the cleaner state $x_{t-1}^{(i)}$ is computed according to the reverse diffusion equation:

$$x_{t-1}^{(i)} = \frac{1}{\sqrt{\alpha_t}} \left(x_t^{(i)} - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta \left(x_t^{(i)}, t, c^{(i)} \right) \right) + \sigma_t \mathbf{z}, \tag{4}$$

where $\mathbf{z} \sim \mathcal{N}(0, \mathbf{I})$ represents random noise.

By repeating this process, the final clean patch $x_0^{(i)}$ is obtained. Finally, all the generated patches $\{x_0^{(i)}\}_{i=1}^{N}$ are aggregated together to reconstruct the complete form of ancient Chinese character. In this step, overlapping areas are averaged to avoid boundary artifacts.

4 Experiments

In this section, we perform extensive experiments to validate the effectiveness of our method. Specifically, we introduce the dataset and the metrics first, which is followed by the introduction of the

experiment setup. Then, we present the quantitative and qualitative results. Finally, we perform ablation experiments.

4.1 Dataset and Metrics

We construct the Font-607 dataset using images from the Guoxue Dashi website[1]. Starting from 11,174 character IDs, we retain characters with forms in all six scripts (OBI, BI, WSS, SS, CS, and RS) and remove variant-character entries with identical forms to prevent duplicate forms across data splits. Font-607 contains 607 characters and 3,642 images, divided by character ID into 427 training and 180 test characters, with all six forms of each character kept in the same subset. Since genuinely lost forms have no ground truth, we simulate their generation on test characters. Source forms provide the conditions, while target forms are used only for evaluation.

For component pre-training, we construct a separate dataset containing 337 groups of corresponding component images across scripts. The components are cropped from character images collected from the website[2]. The source characters used for these crops are distinct from those in Font-607, ensuring that component pre-training does not use its training or test character images.

To evaluate the quality of form generation of ancient Chinese characters, we select the Learned Perceptual Image Patch Similarity (LPIPS) Zhang et al. (2018a), Structural Similarity Index Measure (SSIM) Wang et al. (2004), and cosine similarity as the evaluation metrics. These three metrics evaluate the similarity between the generated character form and the ground truth from various perspectives.

4.2 Experiment Setup

We train Anc-Diff on an NVIDIA GeForce RTX 4090 with the following parameters. The number of steps in the diffusion model is set as $T = 1000$. A linear noise schedule is employed, in which the value of β linearly changes from $\beta_1 = 0.0001$ at the first step to $\beta_T = 0.02$ at the final step. The size of the image is set to 112×112, and the size of the patch is set to 64×64. The model is pre-trained for 300 epochs and trained for 100 epochs. If the pre-training stage is removed, the model is trained for 300 epochs. The training schedules are selected separately for the two settings based on their best performance. The Adam optimizer is employed, with the learning rate set to 0.00002 and the weight decay set to 0.0001.

4.3 Quantitative Evaluation

To comprehensively evaluate our method, we conduct an extensive comparison with various existing methods. Specifically, we compare our method with 7 image-to-image translation methods, including both GAN-based methods (CycleGAN, Pix2pix, DRIT++) and diffusion-based methods (ControlNet, T2I-Adapter, BBDM, OBSD). In addition, we also compare our method with FontDiffuser, a diffusion-based method specifically designed for few-shot font generation.

In this section, we present the experimental results of generating Seal Script and Clerical Script. As presented in the Table 1, our method achieves the best experimental results. In terms of LPIPS, our method achieves the lowest values of 0.2346 and 0.2389 for the tasks of Seal Script generation and Clerical Script generation, respectively, outperforming the best-performing baseline in each task, with corresponding scores of 0.2517 and 0.246. When it comes to cosine similarity, our method achieves the highest similarity values of 0.8094 and 0.8505 for the two tasks, respectively. Specifically, our method surpasses the second-best results of 0.7841 (T2I-Adapter) in Seal script generation and 0.8406 (ControlNet) in Clerical script generation. While our scores in the SSIM are not the highest, they remain highly competitive. In summary, these quantitative results confirm that the ancient Chinese character forms generated by our method have the highest similarity with the ground truth. This superiority is primarily attributed to the effective utilization of component priors and the enhanced guidance of conditional images provided by our method.

[1] https://www.guoxuedashi.com/

[2] https://www.chinese-linguipedia.org/search_source.html

Table 1: Quantitative results. ↑ means the higher the better, ↓ means the lower the better. Best experimental results are marked in bold, and best results for each comparison method are underlined.

	Target Script: SS				Target Script: CS			
Method	Source	CosSim↑	SSIM↑	LPIPS↓	Source	CosSim↑	SSIM↑	LPIPS↓
CycleGAN Zhu et al. (2017)	OBI	0.7305	0.4343	0.3362	OBI	0.719	0.3914	0.3957
	BI	0.7414	0.4343	0.3342	BI	0.7447	0.343	0.3881
	WSS	0.7347	0.4414	0.3248	WSS	0.7558	0.3687	0.3747
	CS	0.7395	0.4235	0.3433	SS	0.7429	0.3365	0.3899
	RS	0.7439	0.4482	0.3109	RS	0.7954	0.3882	0.3121
Pix2pix Isola et al. (2017)	OBI	0.7036	0.4924	0.3174	OBI	0.7515	0.4222	0.3588
	BI	0.6798	0.5009	0.3387	BI	0.7351	0.4201	0.3458
	WSS	0.7234	0.485	0.3097	WSS	0.7411	0.4224	0.3441
	CS	0.6998	0.4873	0.3094	SS	0.7556	0.4245	0.3325
	RS	0.7123	0.4885	0.3002	RS	0.7881	0.4356	0.2948
DRIT++ Lee et al. (2018)	OBI	0.7082	0.3352	0.3695	OBI	0.7098	0.2109	0.4571
	BI	0.7284	0.4336	0.3454	BI	0.6272	0.4668	0.4818
	WSS	0.7019	0.4944	0.3587	WSS	0.7553	0.3753	0.3646
	CS	0.7253	0.4589	0.3534	SS	0.7388	0.3557	0.3941
	RS	0.7367	0.4547	0.318	RS	0.797	0.3829	0.3221
BBDM Li et al. (2023)	OBI	0.7316	0.4943	0.3001	OBI	0.7179	0.4511	0.3983
	BI	0.7217	0.5124	0.3206	BI	0.7706	0.4226	0.3516
	WSS	0.7468	0.486	0.299	WSS	0.7092	0.462	0.3884
	CS	0.7465	0.5046	0.2952	SS	0.7221	**0.4678**	0.3947
	RS	0.7648	**0.5157**	0.2787	RS	0.789	0.4666	0.3003
ControlNet Zhang et al. (2023)	OBI	0.7137	0.4176	0.3242	OBI	0.7546	0.3075	0.3839
	BI	0.7334	0.4396	0.3039	BI	0.7511	0.3103	0.382
	WSS	0.7345	0.4391	0.3065	WSS	0.7608	0.3166	0.3693
	CS	0.7469	0.4486	0.2949	SS	0.7744	0.3311	0.3554
	RS	0.7745	0.4832	0.2534	RS	0.8406	0.4272	0.246
T2I-Adapter Mou et al. (2024)	OBI	0.7377	0.4514	0.2956	OBI	0.7671	0.3064	0.37
	BI	0.7494	0.4601	0.29	BI	0.771	0.3083	0.3664
	WSS	0.7552	0.4631	0.2832	WSS	0.7828	0.3534	0.3417
	CS	0.7711	0.472	0.2624	SS	0.8044	0.397	0.3042
	RS	0.7841	0.4839	0.2517	RS	0.8341	0.4356	0.2575
FontDiffuser Yang et al. (2024)	OBI	0.7039	0.4876	0.3515	OBI	0.7508	0.4449	0.3677
	BI	0.7297	0.4707	0.3422	BI	0.7637	0.4305	0.357
	WSS	0.7262	0.4626	0.3354	WSS	0.7713	0.4221	0.3504
	CS	0.75	0.4587	0.3431	SS	0.7844	0.4261	0.3389
	RS	0.7598	0.4505	0.3175	RS	0.8287	0.4432	0.2778
OBSD Guan et al. (2024)	OBI	0.7314	0.4959	0.3333	OBI	0.7378	0.4058	0.3954
	BI	0.715	0.5015	0.3505	BI	0.7314	0.3996	0.4075
	WSS	0.743	0.4925	0.328	WSS	0.7445	0.4235	0.3859
	CS	0.7557	0.5014	0.3053	SS	0.7783	0.4296	0.326
	RS	0.7707	0.5079	0.2784	RS	0.8184	0.4507	0.2694
Anc-Diff (Ours)	OBI	0.7716	0.4856	0.2802	OBI	0.7835	0.3994	0.3428
	BI	0.7688	0.4943	0.2731	BI	0.7865	0.3994	0.3398
	WSS	0.7703	0.4926	0.2717	WSS	0.7858	0.4187	0.3264
	CS	0.798	0.4998	0.2503	SS	0.8214	0.4404	0.2871
	RS	**0.8094**	0.5064	**0.2346**	RS	**0.8505**	0.4592	**0.2389**

4.4 Qualitative Evaluation

In the preceding experiments, we evaluated all methods by generating Seal script and Clerical script from five different source scripts. To comprehensively evaluate the generation quality, we present the optimal experimental results of each comparison method based on LPIPS, along with the experimental results of our method. The corresponding qualitative results are presented in Figure 4. For clarity, the ground truth and our results are highlighted with red boxes. We present the Seal script generation results here, and those for the Clerical script are provided in the Appendix A.1.

Figure 4 displays the ground truth for 12 Seal script characters as well as the generation results of all evaluated methods. The visual comparisons demonstrate that our Anc-Diff method outperforms existing methods. For instance, Pix2pix produces mode collapse where several generated characters appear nearly identical. Meanwhile, CycleGAN, DRIT++, and FontDiffuser only apply minor dis-



Figure 4: Qualitative results of generating Seal Script.

Table 2: Expert evaluation results. Scores range from 1 to 5. Acc., Sty., Int. and Avg. represent structural accuracy, style similarity, stroke integrity and average score, respectively.

Method	Acc.	Sty.	Int.	Avg.
CycleGAN	2.88	2.50	3.04	2.81
Pix2pix	1.37	1.12	1.33	1.27
DRIT++	2.54	2.33	2.71	2.53
BBDM	1.67	2.08	1.83	1.86
ControlNet	3.06	3.43	3.34	3.28
T2I-Adapter	3.09	3.67	2.89	3.22
FontDiffuser	2.87	3.07	2.83	2.92
OBSD	2.96	3.62	3.21	3.26
Anc-Diff	**3.37**	**4.12**	**3.88**	**3.79**

tortions to the source images, failing to achieve the necessary cross-domain transformation. These results suggest that general GAN-based image-to-image translation methods and standard FFG models struggle with form generation task of the ancient Chinese characters. While BBDM, ControlNet, T2I-Adapter, and OBSD produce results that are relatively similar to the real ancient Chinese characters in terms of style and glyph, they often struggle to capture fine-grained details, which leads to incomplete or broken strokes. In contrast, our Anc-Diff method effectively captures local features and leverages conditional guidance to ensure structural integrity. Consequently, character forms generated via our method exhibit the highest similarity in both script style and content consistency when compared to the ground truth. To assess the statistical significance of the observed improvements, we perform paired tests on evaluation scores for individual characters and report the results in Appendix A.5. These results provide additional evidence supporting the effectiveness of our method.

4.5 Quantitative Human Expert Evaluation

To further validate the generative quality from a professional perspective, we invite 3 experts in ancient Chinese paleography to independently evaluate every character generated by each method across the entire test set. Each character is rated on a five-point scale along three dimensions: structural accuracy, style similarity, and stroke integrity. As shown in Table 2, we report the mean score for each dimension, as well as an overall average score for these three dimensions. Our method achieves the highest mean score in all three dimensions, indicating that its generated characters better preserve the structure, style, and stroke integrity than those produced by the baseline methods.

4.6 Ablation Study

In this section, we conduct a series of ablation experiments to verify the effectiveness of our method. Here, we take the task of generating Seal script as an example.

4.6.1 Effectiveness of Component Pre-training Stage

To evaluate the effectiveness of component pre-training, we compare our method with a variant trained from scratch without the pre-training stage (w/o Pre.). Each variant uses the best-performing training schedule. Table 3 presents the experimental results for generating Seal and Clerical scripts. The results show that the pre-training stage helps the method generate ancient Chinese character forms that are more similar to the ground truth. These results demonstrate the effectiveness of incorporating component pre-training into our framework.

4.6.2 Effectiveness of Feature Injection Positions

To validate the effectiveness of feature injection positions, we conduct an ablation study across various injection positions, including the bottleneck and all upsampling stages (Up4–Up1), and

Table 3: Results of ablation study. SS represents Seal Script and CS represents Clerical Script.

Target	Method	CosSim↑	SSIM↑	LPIPS↓
SS	w/o Pre.	0.8025	0.5062	0.2470
SS	Ours	**0.8094**	**0.5064**	**0.2346**
CS	w/o Pre.	0.8353	0.4493	0.2556
CS	Ours	**0.8505**	**0.4592**	**0.2389**

Table 4: Results of feature injection at different positions.

Method	CosSim↑	SSIM↑	LPIPS↓
Down1	0.8027	0.4981	0.2423
Up4	0.8037	0.5014	0.2427
Up3	0.7842	0.4927	0.2683
Up2	0.7466	0.4441	0.3134
Up1	0.7079	0.4125	0.3772
Bottleneck (Ours)	**0.8094**	**0.5064**	**0.2346**

compare them against the traditional input stage (Down1). The results in Table 4 demonstrate that injecting features at the bottleneck achieves the best performance. The injection at bottleneck allows the conditional features to capture the global layout, acting as a global guide that maintains structural stability throughout the generation. In contrast, early injection at Down1 tends to interfere with the extraction of initial noise features. Similarly, while injection at upsampling layers (Up4-Up1) focus on local details, it lacks the global view of the bottleneck, resulting in structural errors.

4.6.3 Effectiveness of Patch Sampling Module

In our patch sampling module, we combine fixed-position sampling with random-position sampling. To verify the effectiveness of this strategy, we conduct an ablation study to demonstrate the importance of each sampling component. Furthermore, we evaluate different parameters for fixed-position sampling, ranging from 1×1 to 5×5 grids. As shown in Table 5, removing either sampling component degrades performance in terms of CosSim and LPIPS. These results support the effectiveness of combining fixed-position and random-position sampling for character generation. Among the different grid parameters, the 2×2 setup achieves the best results. Overall, these results confirm the effectiveness of our proposed patch sampling strategy.

Table 5: Ablation study of patch sampling module.

	CosSim↑	SSIM↑	LPIPS↓
w/o random	0.7435	**0.5295**	0.3211
w/o fix	0.8041	0.5059	0.2395
1×1	0.7926	0.4884	0.2505
2×2 (Ours)	**0.8094**	0.5064	**0.2346**
3×3	0.8037	0.5053	0.2394
4×4	0.7959	0.5014	0.2416
5×5	0.7948	0.5027	0.2516

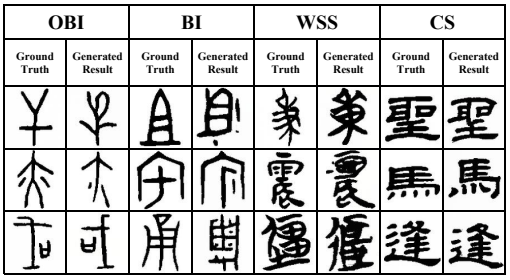

Figure 5: The result of generating different ancient scripts.

4.6.4 Generating Different Character Forms

In order to comprehensively verify the effects of our method across diverse character forms, Figure 5 shows the results of generating different character forms. These results demonstrate the generalization of our method. Quantitative results for generating all earlier scripts are provided in AppendixA.3, further demonstrating the effectiveness of our method.

5 Conclusion

In this paper, we propose a novel diffusion-based method to generate the form of lost ancient Chinese characters. To overcome the lack of domain-specific knowledge, a component pre-training stage is introduced to capture component priors. To enhance guidance in the generation process, a decoupled guidance control module is introduced. By decoupling the conditional feature extraction and injecting features into the U-Net bottleneck, the decoupled guidance control module avoids feature interference and strengthens conditional guidance. In addition, to better capture local information, a unified patch sampling module is used, which ensures the components are accurately generated. Extensive experiments demonstrate that our method improves the generation quality of ancient Chinese characters.

AI USE STATEMENT

In this work, we used generative AI tools to assist with translation and polish writing. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, designing or provide feedback on research methodology or experiments, implementing methods, cleaning and reformating dataset, supporting qualitative and thematic data analysis and interpreting results. Formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refine hypotheses are not applicable to this work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Junbum Cha, Sanghyuk Chun, Gayoung Lee, Bado Lee, Seonghyeon Kim, and Hwalsuk Lee. Few-shot compositional font generation with dual memory. In *Computer Vision – ECCV 2020*, pp. 735–751, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58528-0. doi: 10.1007/978-3-030-58529-7_43.

Xiang Chang, Fei Chao, Changjing Shang, and Qiang Shen. Sundial-gan: A cascade generative adversarial networks framework for deciphering oracle bone inscriptions. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, pp. 1195–1203, New York, NY, USA, 2022.

Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Stargan: Unified generative adversarial networks for multi-domain image-to-image translation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Ruwei Dai, Chenglin Liu, and Baihua Xiao. Chinese character recognition: history, status and prospects. *Frontiers of Computer Science in China*, 1:126–136, 2007.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 8780–8794. Curran Associates, Inc., 2021.

Bin Fu, Fanghua Yu, Anran Liu, Zixuan Wang, Jie Wen, Junjun He, and Yu Qiao. Generate like experts: Multi-stage font generation by incorporating font transfer process into diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024.

Yue Gao, Yuan Guo, Zhouhui Lian, Yingmin Tang, and Jianguo Xiao. Artistic glyph image synthesis via one-stage few-shot learning. *ACM Trans. Graph.*, 38(6), November 2019. ISSN 0730-0301. doi: 10.1145/3355089.3356574.

Haisu Guan, Huanxin Yang, Xinyu Wang, Shengwei Han, Yongge Liu, Lianwen Jin, Xiang Bai, and Yuliang Liu. Deciphering oracle bone language with diffusion models. *CoRR*, abs/2406.00684, 2024.

Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Qiao Yu. Diff-font: Diffusion model for robust one-shot font generation. *Int. J. Comput. Vision*, 132(11):5372–5386, June 2024. ISSN 0920-5691. doi: 10.1007/s11263-024-02137-0.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851. Curran Associates, Inc., 2020.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1952–1961, June 2023.

Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 4296–4304, 2024.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Multiple heads are better than one: Few-shot font generation with multiple localized experts. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 13900–13909, October 2021a.

Song Park, Sanghyuk Chun, Junbum Cha, Bado Lee, and Hyunjung Shim. Few-shot font generation with weakly supervised localized representations, 2021b.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi (eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pp. 234–241, Cham, 2015. Springer International Publishing.

Chitwan Saharia, William Chan, Huiwen Chang, Chris Lee, Jonathan Ho, Tim Salimans, David Fleet, and Mohammad Norouzi. Palette: Image-to-image diffusion models. In *ACM SIGGRAPH 2022 conference proceedings*, pp. 1–10, 2022.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021.

Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861.

Yangchen Xie, Xinyuan Chen, Li Sun, and Yue Lu. Dg-font: Deformable generative networks for unsupervised font generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5130–5140, June 2021.

Zhenhua Yang, Dezhi Peng, Yuxin Kong, Yuyi Zhang, Cong Yao, and Lianwen Jin. Fontdiffuser: one-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. AAAI'24/IAAI'24/EAAI'24. AAAI Press, 2024. ISBN 978-1-57735-887-9. doi: 10.1609/aaai.v38i7.28482.

Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 3836–3847, October 2023.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018a.

Yexun Zhang, Ya Zhang, and Wenbin Cai. Separating style and content for generalized style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018b.

Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22490–22499, June 2023.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

A TECHNICAL APPENDICES AND SUPPLEMENTARY MATERIAL

A.1 QUALITATIVE EVALUATION OF CLERICAL SCRIPT GENERATION

Due to space limitations in the main text, we only present the experimental results of Seal script generation. In this section, we provide additional experimental results for Clerical script generation to offer a more comprehensive evaluation of our method. Unlike the significant glyph differences between Seal and Regular scripts, Clerical script is very similar to the reference Regular script. Since the Clerical script shares high similarity with the Regular script, the generation can be viewed as a simple font style transfer task. Consequently, the overall difficulty of generating Clerical script is relatively lower. This is why some other methods fail on the Seal script but work well on this task. However, even in this less challenging scenario, our method continues to demonstrate superior performance, particularly in preserving fine-grained stroke textures and ensuring visual clarity.

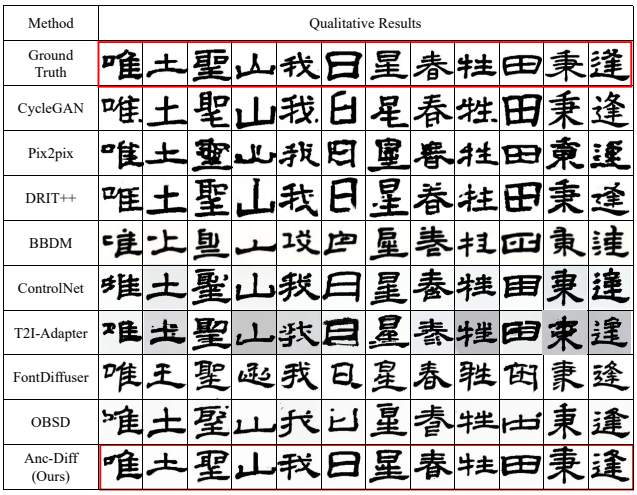

Figure 6: Qualitative results of our method and other comparison methods.

The experimental results are shown in Figure 6. We present the ground truth of 12 Clerical script characters, as well as the generation results of all methods. For this easier task, Pix2pix produces more recognizable characters than in Seal script generation, although some strokes appear distorted or overly thick.

For CycleGAN, some characters stay exactly the same as the original Regular Script. In the case of DRIT++, the generated characters often contain many broken or missing parts. As a font generation method, FontDiffuser performs much better when the differences between the two scripts are small. However, since it mainly relies on slight distortions of the original image, its overall performance is still not good enough when compared to the Clerical script results shown in the ground truth. BBDM performs well in Seal script generation, but it does not show its advantages in this simpler task. ControlNet, T2I-Adapter, and OBSD generate characters that resemble Clerical script, but some strokes differ in shape or placement from the ground truth. Our Anc-Diff method achieves superior performance because it incorporates component priors and enhances guidance of conditional features. By focusing on these character components, the model can better capture the local features of ancient characters. When compared with the ground truth, our method produces the best results in terms of both font style and content similarity.

We also present the human expert evaluation on these results. As shown in Table 6, we report the mean score for each dimension, as well as an overall average score for these three dimensions. Our method is highly recognized by experts across all dimensions, with an average score outperforming the baseline methods. The high scores demonstrate that our method effectively leverages component priors to generate a high-quality form of ancient Chinese characters.

Table 6: Expert evaluation results. Scores range from 1 to 5 (higher is better).

Method	Acc.	Sty.	Int.	Avg.
CycleGAN	4.17	4.00	4.25	4.14
Pix2pix	3.92	4.08	4.25	4.08
DRIT++	4.42	4.50	4.58	4.50
BBDM	3.08	3.67	3.75	3.50
ControlNet	4.08	4.67	4.47	4.41
T2I-Adapter	3.75	4.50	4.00	4.08
FontDiffuser	3.75	4.00	4.17	3.97
OBSD	3.92	4.50	4.17	4.20
Anc-Diff	**4.75**	**4.92**	**4.75**	**4.81**



Figure 7: Generated Seal scripts.



Figure 8: Source Seal scripts.

A.2 Visual Results on the Test Set

To fully display our experimental results, we provide all the generation results of our method for Seal and Clerical scripts on the test set. This allows for a comprehensive evaluation of our model across various character structures.



Figure 9: Generated Clerical scripts.



Figure 10: Source Clerical scripts.

Figures 7 and 9 show the generated Seal and Clerical scripts, while Figures 8 and 10 present the corresponding real images. As shown in these figures, our method achieves good overall performance in generating both scripts. The generated characters are very similar to the real images in terms of both style and content. For simple characters with fewer strokes, our method produces high-quality results with clean structures and precise details. For complex characters with complex layouts, our method still maintains a high degree of similarity in terms of overall appearance. This suggests that our model successfully captures the global structure required for ancient characters, even when dealing with a large number of strokes and complex radical combinations.

A.3 Generating Different Character Forms

We show the results of generating different fonts in Figure 11, and these results are presented in Section 4.6.4. Our method achieves highly satisfactory results in generating Seal and Clerical scripts, with both style similarity and structural accuracy closely matching the ground truth. For earlier scripts, the performance of our method decreases as the historical time interval increases. Due to the complex evolutionary patterns and the difficulty in extracting components of early scripts, such as Oracle Bone inscriptions and Bronze inscriptions, these challenges limit the performance of our method.

For each target script, Table7 reports each method's results using the source script that yields its lowest LPIPS, with CosSim and SSIM evaluated under the same configuration. Our method achieves the highest CosSim across all evaluated target scripts and the lowest LPIPS for WSS, SS, and CS. Overall, these results demonstrate the strong performance of our method across the evaluated target scripts.

Table 7: Results of generating different forms.

Target	Source	Method	CosSim↑	SSIM↑	LPIPS↓
OBI	BI	CycleGAN	0.7217	0.4737	**0.3377**
	RS	Pix2pix	0.6818	**0.5439**	0.351
	BI	DRIT++	0.7029	0.452	0.3545
	BI	BBDM	0.7011	0.5334	0.3606
	BI	ControlNet	0.7133	0.4539	0.3413
	BI	T2I-Adapter	0.722	0.4374	0.3511
	BI	FontDiffuser	0.7144	0.4701	0.3656
	BI	OBSD	0.6907	0.5348	0.3846
	BI	Anc-Diff (Ours)	**0.7223**	0.5107	0.3486
BI	OBI	CycleGAN	0.7203	0.4403	**0.3501**
	RS	Pix2pix	0.6888	**0.4892**	0.3775
	SS	DRIT++	0.6992	0.4415	0.3717
	WSS	BBDM	0.7036	0.4749	0.3765
	OBI	ControlNet	0.6809	0.306	0.4027
	WSS	T2I-Adapter	0.7234	0.3611	0.378
	SS	FontDiffuser	0.7026	0.4876	0.3932
	RS	OBSD	0.7026	0.4525	0.3948
	OBI	Anc-Diff (Ours)	**0.7259**	0.4402	0.3647
WSS	RS	CycleGAN	0.7837	0.3269	0.3355
	RS	Pix2pix	0.738	0.3841	0.3442
	SS	DRIT++	0.7669	0.3509	0.3349
	SS	BBDM	0.7626	0.383	0.3467
	SS	ControlNet	0.7924	0.3022	0.3353
	SS	T2I-Adapter	0.7086	0.3184	0.3841
	RS	FontDiffuser	0.757	**0.4025**	0.3644
	SS	OBSD	0.7672	0.3789	0.3486
	SS	Anc-Diff (Ours)	**0.7936**	0.3577	**0.331**
SS	RS	CycleGAN	0.7439	0.4482	0.3109
	RS	Pix2pix	0.7123	0.4885	0.3002
	RS	DRIT++	0.7367	0.4547	0.318
	RS	BBDM	0.7648	**0.5157**	0.2787
	RS	ControlNet	0.7745	0.4832	0.2534
	RS	T2I-Adapter	0.7841	0.4839	0.2517
	RS	FontDiffuser	0.7598	0.4505	0.3175
	RS	OBSD	0.7707	0.5079	0.2784
	RS	Anc-Diff (Ours)	**0.8094**	0.5064	**0.2346**
CS	RS	CycleGAN	0.7954	0.3882	0.3121
	RS	Pix2pix	0.7881	0.4356	0.2948
	RS	DRIT++	0.797	0.3829	0.3221
	RS	BBDM	0.789	**0.4666**	0.3003
	RS	ControlNet	0.8406	0.4272	0.246
	RS	T2I-Adapter	0.8341	0.4356	0.2575
	RS	FontDiffuser	0.8287	0.4432	0.2778
	RS	OBSD	0.8184	0.4507	0.2694
	RS	Anc-Diff (Ours)	**0.8505**	0.4592	**0.2389**

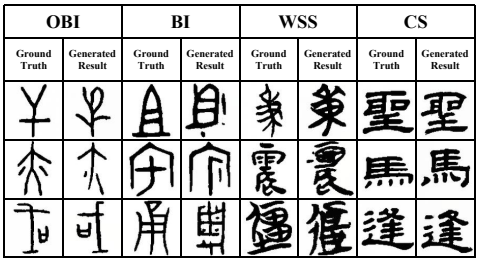

Figure 11: The result of generating different ancient scripts.

A.4 Computational Efficiency Analysis

We use the same model architecture across all three stages. On a single NVIDIA RTX 4090 GPU, the model maintains a constant overhead with 34.8G MACs and a parameter count of 90.1M. Using a batch size of 2, the pre-training stage takes approximately 3.8 hours (13,628.38s), and the training stage requires only about 1.6 hours (5,730.55s). The peak GPU memory usage is approximately 14 GB. The average inference time is approximately 4.2522s per character. These metrics demonstrate that our method is highly feasible for large-scale digitalization projects.

A.5 Statistical Significance Analysis

We conduct statistical significance tests using the same model runs reported in the main text. Specifically, we compute evaluation scores for each of the 180 test characters and pair the scores of our method with those of each baseline by character. We use two-sided paired sign-flip permutation tests with 100,000 resamples and apply Holm correction across all comparisons. The results are shown in Table8:

Table 8: Pairwise significance comparison with our method.

Target	Baseline Method	CosSim (diff ↑)	LPIPS (diff ↓)	SSIM (diff ↑)
SS	**vs. CycleGAN**	$+0.0655\ (p < .001)^*$	$-0.0763\ (p < .001)^*$	$+0.0582\ (p < .001)^*$
	vs. DRIT++	$+0.0727\ (p < .001)^*$	$-0.0834\ (p < .001)^*$	$+0.0517\ (p < .001)^*$
	vs. Pix2pix	$+0.0971\ (p < .001)^*$	$-0.0656\ (p < .001)^*$	$+0.0179\ (p = .005)^*$
	vs. BBDM	$+0.0446\ (p < .001)^*$	$-0.0441\ (p < .001)^*$	$-0.0093\ (p = .075)$
	vs. ControlNet	$+0.0349\ (p < .001)^*$	$-0.0188\ (p < .001)^*$	$+0.0232\ (p < .001)^*$
	vs. T2I-Adapter	$+0.0253\ (p < .001)^*$	$-0.0171\ (p < .001)^*$	$+0.0225\ (p < .001)^*$
	vs. FontDiffuser	$+0.0496\ (p < .001)^*$	$-0.0829\ (p < .001)^*$	$+0.0559\ (p < .001)^*$
	vs. OBSD	$+0.0387\ (p < .001)^*$	$-0.0438\ (p < .001)^*$	$-0.0015\ (p = .703)$

Note: Reported p-values are Holm-adjusted. $^*p < .05$. Positive CosSim/SSIM differences and negative LPIPS differences favor our method.

For Cosine Similarity and LPIPS, our method significantly outperforms all eight baselines (Holm-adjusted $p < 0.001$), providing strong evidence of improved feature-level structural alignment and perceptual quality on the evaluated test set. For SSIM, our method significantly outperforms CycleGAN, DRIT++, Pix2pix, ControlNet, T2I-Adapter, and FontDiffuser, while no statistically significant differences in SSIM are detected relative to BBDM ($p = 0.075$) or OBSD ($p = 0.703$) after Holm correction. Overall, these results demonstrate that our method consistently outperforms the competing methods across most metric-based comparisons on the evaluated test set.