npj | heritage science

Article

https://doi.org/10.1038/s40494-025-02059-1

A structural information-guided cross-modal method for damaged inscription inpainting via vision-language models

Check for updates

Yunjing Liu[1], Erhu Zhang[1,2] ✉, Guangfeng Lin[2] & Jinghong Duan[3]

Restoring inscriptions is crucial for preserving cultural heritage. Current methods primarily focus on visual-level generation and inpainting, ignoring glyph structure information. However, the structural integrity of Chinese characters is frequently compromised in damaged inscription images. To address this challenge, we propose a structural information-guided cross-modal inpainting method. Our dual-branch network includes an inpainting branch and a structure branch. Firstly, to compensate for missing structural information, we pretrain a vision-language model to obtain high-quality glyph structure representations by decomposing each Chinese character into components and structural relationships. Secondly, the glyph structure representation guides the structure branch to optimize features from the damaged character image, producing features that contain more glyph structure information. Thirdly, a feature interaction mechanism injects the optimized features into the inpainting branch, and an adaptive style embedding module improves restoration accuracy in style, structure, and detail. Moreover, a feature sharing module alleviates potential conflicts between branches.

Inscription images play a crucial role in preserving historical knowledge, promoting the art of calligraphy, and safeguarding cultural heritage. However, compared with natural images, inscription images often have a simple background and limited contextual information. The lack of suitable references poses a significant challenge for traditional methods to achieve complete and accurate restoration. Currently, most existing methods rely on generative adversarial networks (GANs)[1–5], which perform restoration by learning unimodal image features. However, these approaches often ignore the inherent structural information of Chinese characters, resulting in redundant or incorrect strokes (as shown in Fig. 1a) as well as missing strokes or disordered components (as shown in Fig. 1b). A critical challenge lies in achieving structurally coherent and accurate restoration when continuous and referable texture information is lacking.

To address these limitations, some researchers have incorporated skeleton priors[6,7] into their models. Although these approaches improve inpainting performance to a certain degree, they still depend on skeleton information extracted from inscription images. Moreover, they require a large and diverse dataset for reference, which limits their applicability in real-world scenarios. To this end, we propose a structure-guided restoration method inspired by human perception of Chinese character composition, which explicitly incorporates glyph structure to enhance inpainting quality under missing information.

Through long-term reading, the brain becomes familiar with the structural relationships inherent in Chinese characters. Even if a character's image is partially damaged, the brain can reconstruct its correct form through structural inference. Inspired by human understanding, we propose that incorporating structural priors into existing restoration models may equip them with analogous inferential capabilities for damaged inscription character inpainting. To achieve this, we have to establish a model to represent the association between the structure of Chinese characters and visual features. Due to the great success of the vision-language model CLIP in text recognition and detection tasks[8–11], we want to use the CLIP model to obtain the Chinese character structural information. For this purpose, we decompose Chinese characters into components and their spatial structure combination relationships, forming ideographic description sequences (IDS) as a source of structural information. Following this line, we pretrain a CLIP model to achieve cross-modal alignment between Chinese character images (visual modality) and their corresponding structural text (i.e., IDS, textual modality). Therefore, the pre-trained CLIP can provide structural priors to compensate for insufficient visual features from damaged inscription character images, which can produce more plausible restorations.

To faithfully reconstruct the original inscription character style, including stroke morphology and thickness, we introduce a style embedding module to enhance the original visual features. In particular, instead of

[1]School of Mechanical and Precision Instrument Engineering, Xi'an University of Technology, Xi'an, China. [2]Department of Information Science, Xi'an University of Technology, Xi'an, China. [3]School of Computer Science and Engineering, Xi'an University of Technology, Xi'an, China. ✉e-mail: eh-zhang@xaut.edu.cn

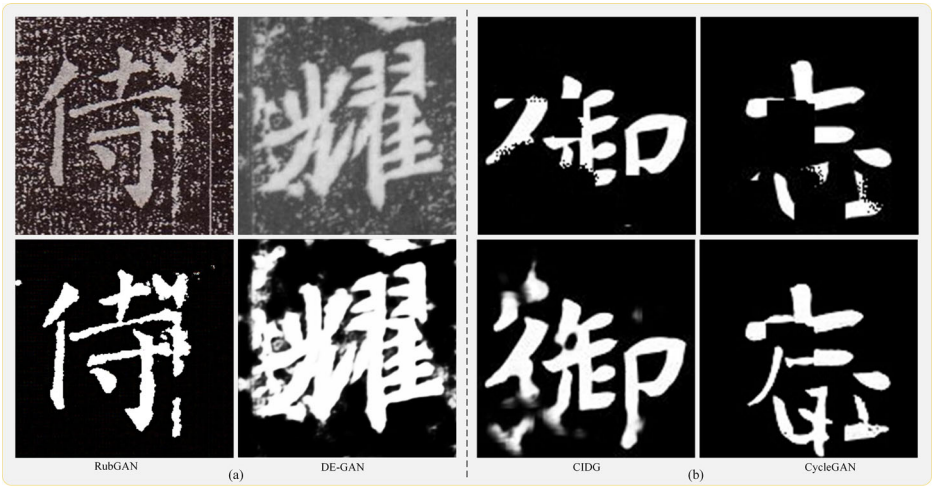

Fig. 1 | Examples of improperly restored images. **a** Preserve meaningless strokes. **b** Disordered structure. The first row displays the original images, while the second row shows the corresponding incorrect inpainting results.

trying to disentangle style and content in Chinese characters, it matches styles via similarity and uses a linear classifier for style prediction and selection. Subsequently, the chosen style features are integrated into the original features to enhance stylistic representation.

Unlike natural images, the glyph structure of inscription images is compromised and they lack the continuous textures or color transitions found in natural images, which lead to insufficient feature information for inscription image restoration. Moreover, the limited inscription image data and the diversity of font styles make it difficult for traditional image restoration methods to be directly transferred and applied to such tasks. Therefore, we propose the glyph structure-guided inpainting network (CINet), which leverages the spatial and structural relationships between Chinese character components for restoration. Specifically, we establish a deep association between the image and structural components using the CLIP model, mapping the structure of the IDS latent space from the CLIP text encoder as prior information, which can guide the Chinese character structural branch (CSB) of CINet to generate a complete glyph structure representation. Additionally, we establish an interaction between the damaged image features and the structural information of Chinese characters, which enhances the inpainting branch (IB) in understanding the character structure. Furthermore, we introduce a style embedding module to maintain consistency in the restoration style.

Our contributions can be summarized as follows:

(1) We propose a learning paradigm for acquiring the structural representation of Chinese characters. By decomposing a Chinese character into a series of components and structural sequences, the CLIP model is employed to perform cross-modal alignment between the inscription image and the structural sequence. Through this alignment, the CLIP text encoder gains the ability to model Chinese character structures, supplies structural priors when the image is compromised.

(2) We propose a dual-branch glyph structure-guided inpainting network (CINet). The two branches collaborate through feature sharing, interaction, and fusion, strengthening the synergy between the two modalities and enhancing the network's sensitivity to glyph structures and restoration performance.

(3) We introduce a style embedding module to enhance the network's sensitivity to different Chinese character styles. Experimental results show that the CINet addresses varying levels of damage and minimizes data dependency, making it particularly suitable for inpainting inscription images.

Methods

Overview of related methods

Image inpainting involves reconstructing damaged regions by utilizing contextual cues and surrounding features. Deep learning has driven the development of numerous image inpainting methods, including local completion based on convolutional neural networks[12], GANs[1], and diffusion models[13,14]. Zhu et al.[14] recently introduced GSDM, a two-stage diffusion framework guided by global structure, which integrates structure prediction and content generation for improved text-image inpainting. Although existing methods have achieved notable progress in content generation, they remain limited in structural modeling, especially in capturing long-range dependencies and complex structural relationships. To address this, recent studies have increasingly adopted self-attention mechanisms to enhance global structure perception. Self-attention, a key component of transformer architecture[15], has proven to be highly effective in modeling global dependencies and has shown substantial success in image inpainting tasks. As a result, there has been growing research[16–18] focused on enhancing transformers to improve reconstruction quality. One notable approach, proposed by Deng et al.[17], is the Tformer network, which leverages Transformer modules. This network adopts a U-Net-like architecture and integrates an innovative linear attention mechanism with a gating module. This design reduces the computational complexity of traditional self-attention while preserving the ability to model long-range dependencies, greatly improving image inpainting and supporting large-scale and real-time tasks. A similar framework, Uformer, proposed by Wang et al.[19], replaces global self-attention with a local enhanced window Transformer module and introduces a learnable multiscale restoration module, which ensures high-quality image restoration details while alleviating the computational burden. To strike a balance between computational complexity and restoration quality, Huang et al.[20] developed a method that enhances Transformer performance in high-resolution image restoration. They designed a cross-channel attention mechanism to model global dependencies, implementing sparse attention distribution by replacing Softmax with ReLU, thus mitigating performance bottlenecks due to high computational complexity. To further enhance the preservation of restoration details, Chen et al.[18] proposed the M × T framework, combining Mamba with Transformer to leverage their synergy for improving detail recovery and ensuring global semantic consistency in image inpainting tasks. In addition to self-attention, researchers have explored various other attention mechanisms to further enhance image inpainting. Cheng et al.[21] introduced a lightweight framework that incorporates an attention module into group convolutions. This model uses a rotation mechanism to assign attention weights between groups, enhancing the interaction between global and local information, making it especially suitable for resource-limited applications. Wang et al.[22] proposed three attention networks aimed at boosting image restoration performance. Chen et al.[23], aiming to process global information more effectively, reduced the resolution of damaged images and used a U-Net-like architecture for global feature capture. They also added a second branch with multiscale channel attention for local restoration and fused the outputs of both branches to

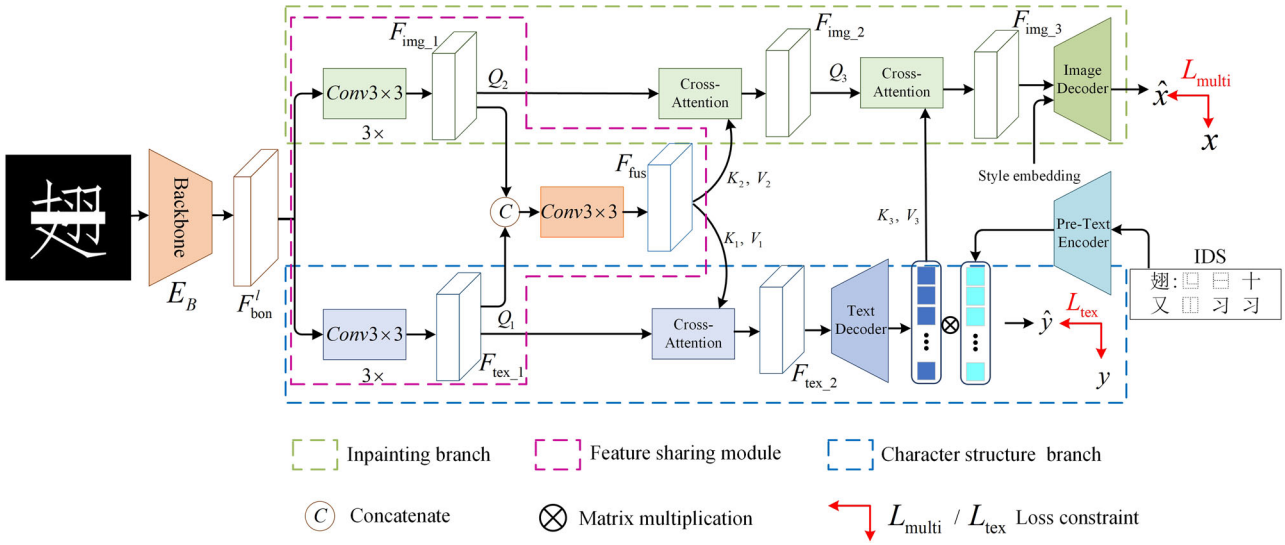

Fig. 2 | Overall framework of CINet. The architecture of CINet, which consists of an inpainting branch and a Chinese structure branch for structural awareness.

improve the final restoration quality. The research above highlights the pivotal role of attention mechanisms in image restoration. By modeling the relationships between global and local features, attention mechanisms effectively leverage contextual information to restore damaged regions. While natural images provide rich information with diverse colors and textures, supporting robust attention mechanisms use, inscription images typically feature simpler backgrounds with fewer usable features, making inpainting more challenging. Therefore, to address the lack of contextual information in inscription images, it is essential to introduce additional reference data during the inpainting process to improve restoration quality.

Due to their age, many crucial glyph structures in inscription images may have been damaged. The goal of inpainting is to restore the complete inscription image, even when structural information is missing. This requires a deep understanding of Chinese character structure and ensures that the restored characters maintain the same style as the original. To provide a comprehensive overview, the following section discusses restoration methods for inscription images, extending the related research on calligraphy and document image restoration. Existing work primarily relies on image features for inpainting. For instance, Sun et al. proposed the RubGAN model[3], which utilizes a dual discriminator design: one focuses on detailed information, while the other captures global features. By working together, these discriminators help the generator produce restoration results with richer details and more coherent structures. Chen et al.[24] enhanced the GAN framework by incorporating dilated convolutions[25] into the generator, expanding the receptive field and improving the model's feature extraction capabilities. While these methods are effective for lightly damaged inscription images, they fall short when dealing with severe damage, as relying solely on image features provides insufficient information, leading to reduced restoration performance. To overcome this limitation, some researchers[6,7,26] have attempted to incorporate additional prior information to improve inpainting performance. Shi et al.[7] proposed a method that uses character skeletons as priors to restore real-world inscription images. Built on a GAN framework, this method leverages multiscale feature fusion to enhance detail restoration. Li et al.[26] introduced a network similar to font style transfer, incorporating template images to provide structural information and using a style encoder for style consistency. Shi et al.[6] proposed a parallel-task framework for denoising inscription images, where image and skeleton features are fused using spatial and channel attention mechanisms and reinjected to preserve glyph structures. Song et al.[27] incorporated a self-attention mechanism into the GAN generator to better capture global information, employing multiple loss functions to improve handwritten Chinese character inpainting. These methods improve restoration accuracy by extracting skeleton information or using attention mechanisms to enhance the utilization of image features. However, they are fundamentally constrained by their reliance on image features, limiting their ability to restore glyph integrity when the image data quality is poor. In addition, variational autoencoders (VAEs)[28], commonly used for image inpainting and reconstruction, encode images into a latent space for progressive reconstruction. Pathak et al. further proposed the context encoder network (CENet)[29], which combines VAEs with generative adversarial networks (GANs) to improve performance. A recent study by Zhao et al.[30] proposed a cross-autoencoder framework for inscription image inpainting, which employs dilated convolutions and channel attention for parallel feature encoding, and uses shared-parameter decoders optimized with multiple loss functions to improve inpainting performance. In related research, Zhang et al.[4] expanded the dataset by modeling noise in calligraphy images and used GANs to remove noise patches. Souibgui et al.[5] proposed a document restoration network based on a conditional GAN with a U-Net architecture, designed to handle watermarks, ink stains, and uneven backgrounds in document images. Lugo-Torres et al.[31] applied a CycleGAN framework[2] to address uneven backgrounds in document images (e.g., stains and creases), improving the readability of the documents. In summary, while existing methods for inscription image inpainting have advanced, relying solely on image features is insufficient for restoring glyph integrity and accuracy. Therefore, integrating glyph structure information into restoration networks is essential for improving the performance of inpainting methods, especially when dealing with severely damaged inscription images.

Overall architecture of the proposed method

To enhance the model's ability to extract discriminative features in severely damaged scenarios, we propose the CINet, a cross-modal glyph structure-guided inpainting network. As illustrated in Fig. 2, the CINet consists of a backbone network (E_B), an inpainting branch (IB), a Chinese character structural branch (CSB), and a pretrained text encoder (E_{TEX}). The CINet integrates the character structure information from the CSB into the IB through a cross-attention mechanism, allowing the IB to focus on damaged areas and thereby achieving more accurate restoration results. E_{TEX} is derived from a pretrained CLIP model on large-scale data. The structural vector served as a cross-modal structural prior, learned from extensive the pretraining data rather than the limited samples of the current inpainting task. Moreover, this prior has already been well aligned with the visual features of complete Chinese characters during pretraining and

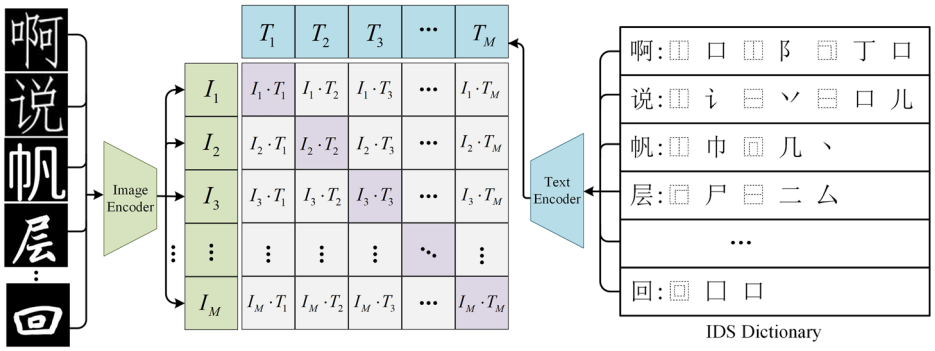

Fig. 3 | CLIP model for Chinese character recognition. The model comprises two encoders: an image encoder for character images and a text encoder for ideographic description sequences (IDS).

demonstrates strong invariance to font style variations. As a result, even with a limited number of training examples, CINet can reliably obtain accurate structural information via the CSB branch, thereby significantly improving the feature representation and inpainting performance of the IB branch. To clarify the explanation of this process, we formalize it as follows.

$$\hat{x} = IB\left(E_{\mathrm{B}}\left(x_n\right), CSB\left(x_n\right), S_{\mathrm{emb}}\right) \tag{1}$$

where x_n represents the damaged image, while IB, E_{B}, and CSB denote the functions describing the IB, $\mathrm{E_B}$, and CSB, respectively. S_{emb} represents the style embedding.

Pre-training CLIP for glyph structure representation

Chinese characters have significant structural characteristics, and their glyphs are composed of multiple components arranged according to specific spatial relationships (such as left and right, up and down). For example, the character "构" consists of the components "木", "勹", and "厶", arranged in a left-right and semi-enclosing structure. To formalize the structural representation of Chinese characters, the Unicode standard defines IDS. IDS consists of structural symbols (e.g., ⿰ and ⿸ to represent structures) and component symbols (e.g., 木, 勹, and 厶 to represent components) defined by Unicode. It encodes Chinese character components and their hierarchical relationships, thereby achieving the standardized decomposition of glyph structures.

For inscription image inpainting, although different fonts exhibit significant visual variations, their fundamental structures typically remain consistent. However, relying solely on visual features makes models susceptible to style interference, hindering the learning of unified and stable structural representations. To address this, we introduce a structure-aware cross-modal alignment mechanism inspired by CLIP. Although CLIP is effective in semantic tasks such as retrieval and classification, it lacks structural modeling capabilities, limiting its suitability for structure-sensitive tasks involving Chinese characters. To improve structural understanding, we propose an alignment paradigm that replaces natural language with sequential representations of Chinese character structures. These sequences are then aligned with character images during training, guiding the model to learn style-independent structural representations and improving its ability to model glyph structures. In particular, when training the CLIP model, we input Chinese character images and ideographic description sequences (IDS) into the image encoder and text encoder, respectively, as shown in Fig. 3. Specifically, we adopt the ResNet-50 architecture as the image encoder to extract the image feature vector (I), and use a Transformer-based text encoder that models sequential dependencies and projects its output through a linear layer to match the dimensionality of I, resulting in the text feature vector (T).

We apply contrastive loss optimization to align a batch of M image feature vectors and text vectors. The specific loss function is as follows, where the first term represents the image-to-text loss, while the second term represents the text-to-image loss.

$$L_{\mathrm{clip}} = -\frac{1}{2M}\left(\sum_{i=1}^{M} log\frac{exp\left(I_i \cdot T_i\right)}{\sum_{j=1}^{M} exp\left(I_i \cdot T_j\right)} + \sum_{j=1}^{M} log\frac{exp\left(T_j \cdot I_j\right)}{\sum_{i=1}^{M} exp\left(T_j \cdot I_i\right)}\right) \tag{2}$$

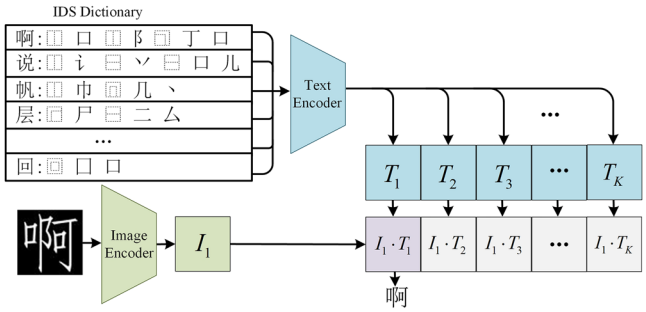

Fig. 4 | The inference process of Chinese character recognition. Chinese character categories are determined by computing the cosine similarity between the output vectors of the image and text encoders.

By training on large-scale image–IDS pairs, the CLIP model learns to align the visual features of complete Chinese characters with their structural representations. This alignment can establish a stable structural prior after training. On the one hand, the representation of IDS sequences remains invariant to font style variations and exhibits strong consistency. On the other hand, it is independent of the data scale in downstream restoration tasks. Consequently, even when training data is limited, the model can rely on the structural vector produced by the text encoder to provide high-quality structural guidance for the inpainting network.

After training, the inference process is illustrated in Fig. 4. The CLIP model first encodes the Chinese character image using the image encoder to obtain its image feature vector (I). Simultaneously, the K predefined candidate Chinese characters are decomposed into IDS and encoded by the $\mathrm{E_{TEX}}$ into a series of textual feature vectors $T = \{T_1, T_2, \ldots, T_K\}$. By computing the similarity between the image and textual representations, the model outputs the index k^* corresponding to the character category with the highest similarity. Accordingly, the corresponding textural feature T_k represents the structural information of the Chinese character. To clarify, we provide the following formula.

$$k^* = \underset{k\in\{1,2\ldots,K\}}{\arg\max}\, S\left(I, T_k\right) \tag{3}$$

where S denotes the cosine similarity calculation.

Backbone network

ResNet[32], renowned for its innovative residual connection design, excels in both global and local feature extraction and is widely used in visual tasks[33,34].

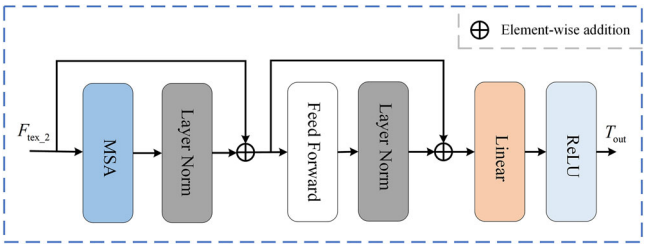

Fig. 5 | Text decoder of the Chinese character structure branch. It generates the structural vector (T_{out}) of Chinese characters.

Therefore, we select ResNet-50 as the backbone for the backbone feature extraction.

$$F_{bon} = \mathrm{ResNet}_{50}(x_n) \tag{4}$$

where F_{bon} represents the extracted the backbone features.

Next, we respectively use three 3×3 convolutions to obtain the initial feature F_{img_1} of the inpainting branch and the initial feature F_{tex_1} of the structural branch.

Feature sharing module

Benefiting from the introduction of the CSB, the IB becomes more sensitive to glyph perception, thereby significantly improving the inpainting results. Notably, the IB focuses on restoring pixel-level details, while the CSB emphasizes the semantic features of the overall glyph structure. To align the features with the needs of different tasks, we design a feature sharing module (FSM) to address potential inconsistencies, as shown by the pink dashed box in Fig. 2. First, the FSM applies three convolutional layers (with a kernel size of 3×3) to each task branch to extract task-specific features. Then, the extracted features are concatenated using the concatenation operation (Cat). Finally, a 3×3 convolutional layer is applied to the concatenated features to fuse them, resulting in the sharing feature F_{fus} as shown in Eq. (5), which can enhance the interaction and correlation between the initial features of the CSB and IB branches.

$$F_{fus} = Conv_{3\times3}\left(Cat\left(F_{img\_1}, F_{tex\_1}\right)\right) \tag{5}$$

Character structure branch

The CSB, consisting of a cross-attention module (CA_{TEX}) and a text decoder (D_{TEX}) as shown in Fig. 2, is responsible for extracting high-quality glyph structure information, helping the IB more sensitively perceive the semantic features of the glyphs during restoration process. Therefore, obtaining and utilizing cross-modal glyph structure information is a critical task. Inspired by the cross-modal learning between images and texts in the CLIP model, we first pretrained a CLIP model for Chinese character image recognition. Then, we applied the E_{TEX} from the pretrained CLIP model to get Chinese character structure representations for constraining the output features of the CSB. This process forces the CSB to extract key glyph information from the damaged images. Finally, the glyph representation is passed to the IB through a cross-attention mechanism, enabling the IB to incorporate glyph constraints and ensure the accuracy of the restored character structure.

In the CSB, the process of glyph structure extraction is as follows. First, the CA_{TEX} is used to obtain an enhanced feature F_{tex_2} from the initial feature F_{tex_1} and the sharing feature F_{fus} as shown in Eq. (6). Then, the D_{TEX} is applied to map the enhanced feature F_{tex_2} to Chinese character structure feature (T_{out}). Finally, the similarity between the output feature of E_{TEX} and T_{out} is calculated to obtain structural information.

$$F_{tex\_2} = CA_{TEX}(Q_1, K_1, V_1) = Softmax\left(\frac{Q_1 K_1^T}{\sigma}\right) V_1 \tag{6}$$

where $Q_1 = W_1^Q F_{tex\_1}$, $K_1 = W_1^K F_{fus}$, $V_1 = W_1^V F_{fus}$, with W_1^Q, W_1^K and W_1^V denoting learnable weight matrices, and σ is the scaling factor.

Since the CSB relies on features from the damaged character image, it encounters limitations of accurately capturing the complete structural information. To improve the representation of Chinese character structures, we leverage the E_{TEX} output from the pre-trained CLIP model as structural prior knowledge, which constrains the prediction vector from D_{TEX} and guides the CSB branch toward generating a more complete and accurate glyph structure. The D_{TEX} is based on a Transformer architecture[15], as illustrated in Fig. 5. Specifically, F_{tex_2} is first processed by a multihead self-attention mechanism (MSA), followed by a feed forward network and a normalization layer. Finally, a fully connected layer outputs the Chinese character structure prediction vector (T_{out}).

The loss function L_{tex} of the CSB consists of two components: the cross-entropy loss L_{rec} and the mean squared error loss L_{dis}. Specifically, we utilize the E_{out} provided by E_{TEX} from CLIP as the target. By optimizing the similarity between the T_{out} from D_{TEX} and E_{out}, we aim to bring T_{out} and E_{out} closer within the feature space. Moreover, we apply L_{dis} to constrain the distance between the T_{out} and E_{out}. The detailed formulas (7)–(9) are as follows:

$$L_{rec} = -\frac{1}{N}\sum_{n=1}^{N} \log\left(\frac{exp\left(T_{out}^n \cdot E_{out}^{y_n}\right)}{\sum_{j=1}^{C} exp\left(T_{out}^n \cdot E_{out}^j\right)}\right) \tag{7}$$

$$L_{dis} = \frac{1}{N}\sum_{n=1}^{N} ||E_{out}^n - T_{out}^n||_2^2 \tag{8}$$

$$L_{tex} = L_{rec} + 0.01 \cdot L_{dis} \tag{9}$$

where N represents the number of samples in a batch. C indicates the number of Chinese character categories, while $E_{out}^{y_n}$ refers to the feature vector corresponding to the ground-truth character label y_n.

Inpainting branch

The inpainting branch (IB) is responsible for restoring the structure, style, and details of Chinese characters. It comprises an image decoder (D_{IMG}), two cross-attention mechanisms (CA_{IMG} and CA_{IMG_TEX}), and a style embedding module as shown in Fig. 2. First, the IB branch performs a feature interaction between the initial feature F_{img_1} and the sharing feature F_{fus} through the cross-attention (CA_{IMG}), obtaining an enhanced feature representation F_{img_2} as shown in Eq. (10). Then, the CA_{IMG_TEX} is employed to inject the cross-modal glyph structure feature extracted from the CSB into the IB, helping it focus on the key parts of the restoration and ensuring the accuracy of the restored Chinese character structure. The feature F_{img_3} generated by the CA_{IMG_TEX} is expressed by Eq. (11).

$$F_{img\_2} = Softmax\left(\frac{Q_2 {K_2}^T}{\sigma}\right) V_2 \tag{10}$$

where $Q_2 = W_2^Q F_{img\_1}$, $K_2 = W_2^K F_{fus}$, $V_2 = W_2^V F_{fus}$, with W_2^Q, W_2^K and W_2^V denoting learnable weight matrices.

$$F_{img\_3} = Softmax\left(\frac{Q_3 {K_3}^T}{\sigma}\right) V_3 \tag{11}$$

where $Q_3 = W_3^Q F_{img\_2}$, $K_3 = W_3^K T_{out}$, $V_3 = W_3^V T_{out}$, withW_3^Q, W_3^K and W_3^V representing learnable weight matrices.

Due to the difficulty of decoupling the style and content features of Chinese characters from the latent feature (i.e., F_{img_3}), we don't adopt the decoupling approach[35,36]. Instead, we use the style embedding to enhance the latent feature representation, where style loss optimization encourages the style embedding module to learn more discriminative style representations, as shown in Fig. 6. Specifically, we assign a unique index to each style category and convert the style index into a style embedding matrix (I_{emb}) through an embedding layer. We then compute the similarity by performing a dot product between I_{emb} and the transformed image features F'_{img_3} from the original features F_{img_3}. The formula is as follows:

$$S_{dot}\left(I_{emb}, F'_{img\_3}\right) = I_{emb} \cdot \left(AvgPool\left(Sigmoid\left(F_{img\_3}\right)\right)\right) \quad (12)$$

The dot product S_{dot} is a widely used and effective similarity measure in deep learning. It is adopted in the self-attention mechanism of Transformer models[15], as well as in various tasks such as sentence transformation and sentiment intensity modeling[37]. Finally, the computed similarity is passed through a fully connected layer to predict the font style, as follows:

$$I_{style} = FC\left(S_{dot}\left(I_{emb}, F'_{img\_3}\right)\right) \quad (13)$$

where I_{style} corresponds to the logits output by the model, and $FC(\cdot)$ represents the output of the fully connected layer.

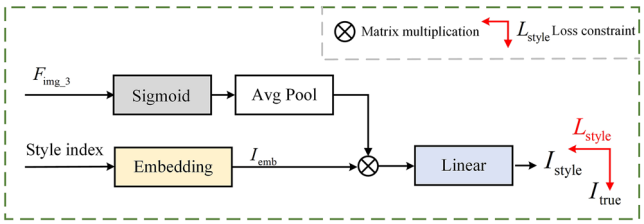

Fig. 6 | Structure of the style embedding module. It is used to capture the style representation of the image.

We apply the multiclass cross-entropy loss L_{style} to constrain the style prediction, as shown in the formula (14):

$$L_{style} = -\frac{1}{N}\sum_{n=1}^{N} log\left(\frac{exp\left(I^n_{style} \cdot I^{t_n}_{true}\right)}{\sum_{j=1}^{C'} exp\left(I^n_{style} \cdot I^j_{true}\right)}\right) \quad (14)$$

where the numerator represents the similarity score between the predicted style feature and the reference vector of its ground-truth category t^n, and the denominator is the sum of similarity scores between the predicted style feature and the style vectors of all style categories C'.

The D_{IMG} is a hierarchical feature fusion network, and its key design is the multilevel feature fusion module (MFM), inspired by ref. 38. MFM integrates both standard convolution and dilated convolution, effectively extracting multiscale features, as shown in Fig. 7. The D_{IMG} begins feature fusion from the deepest layers and progressively fuses toward the shallower layers. After each fusion level, the predicted results are resized to the original dimensions through a 3×3 convolution and an interpolation operation. This design of multiscale supervision can significantly enhance the restoration performance. The decoding process is represented as follows:

$$\hat{x}_0 = \left\{Conv_{3\times3}\left(MFM_l\left(F^l_{bon}, \left(F_{img\_3} + S_{emb}\right)\right)\right) | l = 3, 2, 1, 0\right\} \quad (13)$$

where $\hat{x}_0$ represents the final restoration result. S_{emb} is obtained by transforming the dimensions of I_{emb}.

We employ a multiscale supervision strategy to optimize the restoration process, applying L_1 loss on the outputs of each intermediate layer. The final loss function L_{multi} can be expressed as follows:

$$L_{multi} = \sum_{l=0}^{3} w_l ||\hat{x}_l - x||_1 \quad (16)$$

where $\hat{x}_1$, $\hat{x}_2$, and $\hat{x}_3$ are the outputs of the intermediate layers.

In summary, the overall loss L_{total} for training CINet consists of three components, defined as follows:

$$L_{total} = L_{tex} + L_{style} + L_{multi} \quad (17)$$

Results

Datasets and experimental settings

Due to the lack of publicly available inscription image datasets, we constructed a real-damaged inscription dataset (DID) by collecting images from

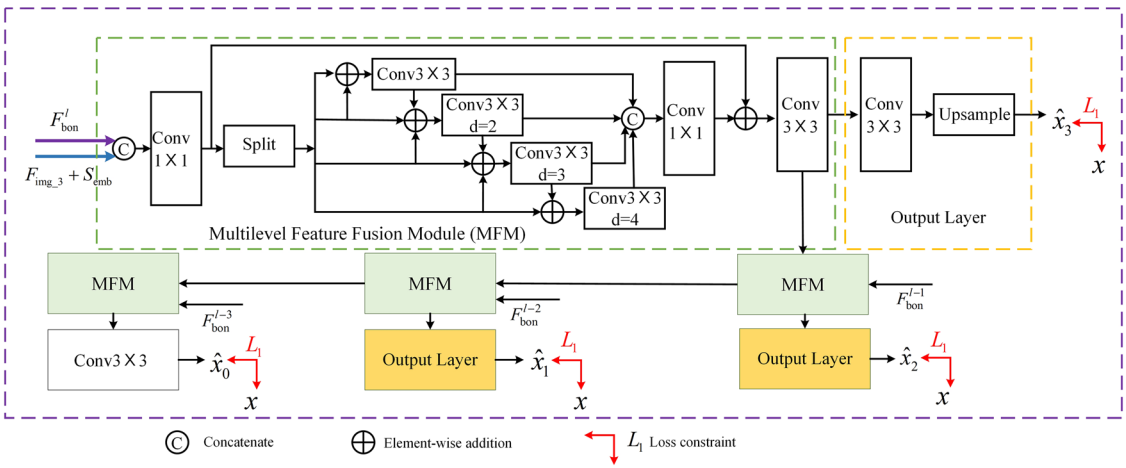

Fig. 7 | The image decoder (DIMG) of the inpainting branch. The decoder mainly comprises multiple feature fusion modules (MFM) and an output layer, producing the final restored image.

a public database[1]. However, the types of damaged inscription images in this dataset are relatively limited and do not cover more severe degradation that may occur in practical scenarios. In addition, the dataset is relatively small, making it insufficient to evaluate the model's generalization ability. To address this issue, we further constructed two synthetic datasets, namely the Damaged Printed Chinese Characters Dataset (DPCCD) and the Damaged Handwritten Chinese Characters Dataset (DHWD), to assist in analyzing the model's performance under various conditions. Furthermore, we introduced a Printed Chinese Characters Dataset (PCCD) to train the CLIP model.

We gathered inscription images from different dynasties to construct the DID, resulting in a dataset of 3295 inscription images, as shown in Fig. 8a. We use 2608 images for training and 687 for testing.

To construct the DPCCD, we collected 10 unseen styles (not part of PCCD) to generate images for 3755 Chinese character categories, totaling 37,550 images. We introduced various damaged types and adjusted the damaged areas to create diverse damaged images, as shown in Fig. 8b. Of which, 31,920 images are used for training and 5630 for testing.

We collected 37,423 images of handwritten Chinese characters from 10 authors from HWDB1.1[39], forming the DHWD. We then generated damaged images by applying various damage types and adjusting the damage areas. A total of 31,841 images are used for training and 5582 for testing.

We collected 120 styles and 3755 Chinese character categories (GB2312 level-1 characters) from a public database[2] to construct the PCCD, resulting in 450,600 Chinese character images. The training set includes 110 styles, with 100 styles randomly selected to generate samples for 1126 Chinese character categories. Additionally, 2629 Chinese character categories are covered by all 110 styles. The total number of training samples amounts to 401,790 images. The test set contains 10 unseen styles, with 11,260 Chinese character images generated as test samples.

We use several evaluation metrics to assess model performance, where the PSNR and SSIM are used to evaluate the pixel-level difference and structural integrity, the LPIPS measures the perceptual difference, and the FID quantifies the difference in data distribution between the reconstructed and ground-truth image. To evaluate the model's ability of restoring styles, we train a style classifier on the test set and use it to calculate style scores (StSc)[40] for the inpainting images.

The model is implemented using the PyTorch framework and trained on an i9-14900KF processor with an NVIDIA GeForce RTX 4090D (24 GB) GPU. The IB uses the Adam optimizer[41] with a learning rate of 2×10^{-4}, while the CSB uses the Adadelta optimizer[42] with a learning rate of 1. Specifically, the DPCCD and DHWD use 128×128 with a batch size of 32, while DID uses 256×256 with a batch size of 8. The coefficients w_l in the loss function L_{multi} are set to [0.5,0.3,0.2,0.1]. To provide a more detailed description of the training process, we present the implementation pseudo code in Algorithm 1.

Algorithm 1. Pseudo code of CINet method

Input: training dataset is $D_{\text{data}} = \{x_i, y_i\}_{i=1}^{N}$, batch size N is 8 and the epoch is 200

Output: inpainting image $\hat{x}$

1. Randomly initialize the model parameter θ.
2. For $i = 1$ to epoch do
3. $\{x_N, y_N\} \leftarrow \text{Sample}(D_{\text{data}}, N)$.
4. $F_{\text{bon}} \leftarrow E_{\text{B}}(x_N)$
5. $F_{\text{fus}}, F_{\text{img\_1}}, F_{\text{tex\_1}} \leftarrow FIM(F_{\text{bon}})$
6. $F_{\text{tex\_2}} \leftarrow CA_{\text{TEX}}(W_1^Q F_{\text{tex\_1}}, W_1^K F_{\text{fus}}, W_1^V F_{\text{fus}})$, $F_{\text{img\_2}} \leftarrow CA_{\text{IMG}}(W_2^Q F_{\text{img\_1}}, W_2^K F_{\text{fus}}, W_2^V F_{\text{fus}})$
7. $T_{\text{out}} \leftarrow D_{\text{TEX}}(F_{\text{tex\_2}}), E_{\text{out}} \leftarrow E_{\text{TEX}}(y_N)$
8. $F_{\text{img\_3}} \leftarrow CA_{\text{IMG\_TEX}}(W_3^Q F_{\text{img\_2}}, W_3^K T_{\text{out}}, W_3^V T_{\text{out}})$
9. $\hat{x} \leftarrow D_{\text{img}}(F_{\text{bon}}, F_{\text{img\_3}}, S_{\text{emb}}), \hat{y} \leftarrow T_{\text{out}} \times E_{\text{out}}$
10. Update CSB
11. Update IB
12. end for

1 https://www.9610.com/index.htm 2https://www.foundertype.com/index.php/FindFont/index

Comparison to state-of-the-art methods

In the inpainting of damaged Chinese characters, the absence of strokes and components significantly hinders the recovery of complete structural information when relying solely on the degraded image. To address this limitation, we utilize the known target character category during the restoration process and employ a trained structural feature extraction model

Fig. 8 | Examples of real and synthetic datasets. a Examples of real-inscription dataset. **b** Examples of the synthetic dataset with various corrosion and damage shapes.

Table 1 | Recognition accuracy of 10 unseen styles (%)

Style Code	FZZDXJW	FZZhangMLBKSJW	FZZhaoMFKSJW	FZZhaoMFXSJW	FZZHengWGBKSJW
Accuracy	99.911%	98.757%	93.517%	82.416%	99.023%
Style Code	FZZhengYJW-R	FZZhenSYJW	FZZhiATJW	FZZHJW	XinhuaNewsType-Bold
Accuracy	99.911	99.911	98.313%	99.911%	99.911%

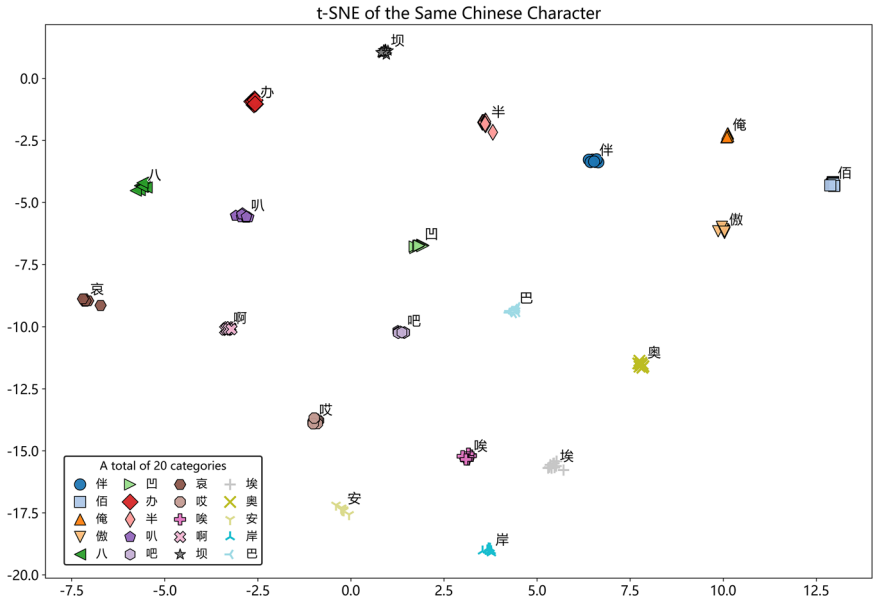

Fig. 9 | t-SNE visualization of feature distributions for the same Chinese character. We use different colors and shapes to distinguish Chinese character classes. Same-class samples cluster tightly in the feature space, indicating robust structural representations by the model.

to obtain a complete structural representation of the character. This representation is subsequently fused with image features to guide the inpainting of missing regions. Unlike conventional cross-style recognition methods, our approach focuses on extracting stable and precise structural features rather than depending on the model's generalization ability to unseen fonts. To ensure the reliability of structural representations, training data should feature clear strokes, standardized structures, and high legibility. The PCCD dataset, composed of high-quality printed Chinese characters with complete and regular structures, provides ideal structural templates for the model. In contrast, handwritten or inscription-based datasets often exhibit considerable structural distortion and noise, making them less suitable for learning stable structural features and potentially degrading restoration performance.

Although the core of our method lies in leveraging stable structural features, we also evaluate the CLIP model across various font styles to gain a more comprehensive understanding of its performance. Specifically, we train the CLIP model using the PCCD, and the pretrained text encoder generates high-quality Chinese character glyph structure representations that are independent of character style. To evaluate the robustness of the CLIP model in recognizing Chinese characters across different font styles, we conduct tests on 1126 Chinese character categories and 10 font styles (both seen and unseen). Notably, there is no overlap between these test sets and the training set. The average recognition rate for the 10 seen font styles is 99.876%, while the rate for the 10 unseen font styles is 97.158% (as shown in Table 1), indicating only a slight drop in performance. Among the unseen fonts, one style achieves a recognition accuracy of approximately 82%, while all others exceed 90%. Overall, the model demonstrates strong robustness to unseen font styles. Even in extreme cases where generalization is limited, reliable structural information can still be retrieved during the inpainting process by leveraging the known character category.

In damaged character scenarios, we rely on the known character category to obtain a complete structural representation from the pretrained model. Accordingly, the model must be capable of extracting consistent structural features across different font styles. To validate this capability, we select the same character rendered in multiple font styles, extract its structural features using the trained model, and project the resulting features into a two-dimensional space using t-SNE, as illustrated in Fig. 9. The t-SNE visualization shows that different font styles of the same character form tight clusters in the feature space, indicating that the structural representations extracted by the model remain stable and consistent across font variations. This property ensures that accurate structural information can still be retrieved from the model to support the inpainting process, even when the input image is damaged, provided that the character category is known.

Because inscription image inpainting is a small-sample problem, we evaluate the performance of various methods under reduced data conditions from two perspectives: the insufficient number of glyph instance samples (G) and style samples (S). We analyze the applicability of different models in small-sample scenarios.

First, reducing G refers to decreasing the number of damaged images with diverse styles under the same IDS conditions. Table 2 summarizes the performance of different methods as G decreases by a step of 2, from 5 to 1 (i.e., $5 \rightarrow 3 \rightarrow 1$), keeping S constant (in this case, S = 2629). As shown in Table 2, the evaluation metrics for all methods generally exhibit a downward trend as G decreases, indicating that glyph structural features are crucial to restoration performance. With only one instance, other models struggle to learn sufficient structural features, resulting in suboptimal restoration. In contrast, CINet performs the best in this scenario, demonstrating its ability to effectively compensate for the lack of information caused by a single glyph sample and showing better adaptability to scarce glyphs. Specifically, compared to the second-place methods, our model outperforms CENet by 0.4442 dB in PSNR; outperforms GSDM by 0.0046 and 0.0205 in SSIM and StSc, respectively; and reduces LPIPS and FID by 0.0051 and 0.6358, respectively.

Second, the reduction of S refers to the decrease in degraded images with identical styles but different IDS. Table 3 reports the performance of various methods when G is 5, and the number of S decreases in an approximately

Table 2 | Impact of reducing glyph instances on performance of the DPCDD Dataset

G of seen inpainting character		PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSC ↑
G = 5	CIDG[4]	18.7166	0.9289	0.0742	11.0120	0.9041
	DE-GAN[5]	18.0239	0.7739	0.1572	95.0079	0.6941
	CycleGAN[2]	13.9018	0.8078	0.1583	45.0151	0.7137
	RubGAN[3]	14.1066	0.8342	0.1336	20.7753	0.7698
	FD-Net[25]	18.4013	0.8448	0.1352	63.6833	0.8092
	RCRN[7]	19.9264	0.9300	0.0647	8.9237	0.9218
	Charformer[6]	19.3447	0.9267	0.0702	13.5724	0.9133
	Uformer[19]	18.2975	0.9129	0.0885	16.1491	0.8560
	Tformer[17]	19.0662	0.9020	0.0635	5.0784	0.9140
	CENet[29]	21.4248	0.9050	0.0836	34.0152	0.9126
	GSDM[14]	21.8589	**0.9508**	0.0421	2.9016	0.9560
	CINet(ours)	**22.5642**	0.9501	**0.0365**	**2.5185**	**0.9728**
G = 3	CIDG[4]	18.6069	0.9300	0.0766	10.7084	0.8995
	DE-GAN[5]	17.9095	0.7792	0.1579	100.5968	0.7162
	CycleGAN[2]	13.6177	0.8071	0.1609	27.9348	0.6124
	RubGAN[3]	14.2907	0.8260	0.1366	27.4403	0.7593
	FD-Net[25]	17.7863	0.8308	0.1539	73.4424	0.7723
	RCRN[7]	19.6830	0.9286	0.0671	9.5869	0.9224
	Charformer[6]	19.6244	0.9184	0.0726	13.0601	0.9140
	Uformer[19]	18.1853	0.9109	0.0906	16.3983	0.8451
	Tformer[17]	18.7107	0.9197	0.0657	5.7012	0.9057
	CENet[29]	21.1021	0.8967	0.0879	35.9235	0.8927
	GSDM[14]	21.5331	0.9485	0.0442	3.1541	0.9488
	CINet(ours)	**22.1492**	**0.9525**	**0.0387**	**2.6915**	**0.9680**
G = 1	CIDG[4]	18.4636	0.9276	0.0781	10.9620	0.9060
	DE-GAN[5]	17.8112	0.7765	0.1623	96.9172	0.6861
	CycleGAN[2]	11.4235	0.7148	0.2517	80.3395	0.5531
	RubGAN[3]	13.3369	0.8087	0.1519	35.3300	0.7259
	FD-Net[25]	14.4154	0.7813	0.2068	118.5517	0.5700
	RCRN[7]	19.5665	0.9215	0.0704	11.5964	0.9156
	Charformer[6]	16.5597	0.7348	0.1592	48.1134	0.7972
	Uformer[19]	17.9328	0.9027	0.0957	19.7934	0.8398
	Tformer[17]	18.8557	0.8502	0.0669	6.0376	0.9110
	CENet[29]	21.3248	0.8957	0.0895	46.9806	0.8909
	GSDM[14]	21.2570	0.9458	0.0464	3.3881	0.9472
	CINet(ours)	**21.7690**	**0.9504**	**0.0413**	**2.7523**	**0.9677**

The optimal results are shown in bold, and the sub-optimal results are underlined. G denotes glyph instance samples.

halving pattern, from 1315 to 329 (i.e., 1315 → 657 → 329). As shown in Table 3, with the reduction of S, the evaluation metrics for all methods generally show a downward trend, indicating that style information also influences restoration performance. However, our model, CINet, still maintains optimal performance compared to all other methods. When S = 329, although our method is 0.0167 lower than the best-performing method in SSIM, it achieves a 0.0823 dB and 0.0202 increase in PSNR and StSc over the second-best, along with reductions of 0.0010 and 0.0700 in LPIPS and FID, respectively. This shows that even with insufficient style information, our method can maintain excellent performance. As shown in Fig. 10, CINet can effectively restores the image, while other models (e.g., DE-GAN, FD-Net, RubGAN, and CENet) produce artifacts. Moreover, compared to models such as RCRN, Tformer, and GSDM, CINet demonstrates greater precision in restoring glyph structures (as highlighted by the red boxes), while better preserving details and avoiding structural blurring or loss.

To more clearly and intuitively demonstrate the performance of CINet under limited G and S samples, we present Table 4 and Fig. 11. Table 4 shows the inpainting effects of CINet as G and S decrease. It is evident that even with a minimal number of glyph categories and style samples (G = 1, S = 329), our model still outperforms most other models tested at G = 5 and S = 2629 (as shown in Table 2). This proves the effectiveness and applicability of our method under small-scale conditions. Figure 11a illustrates the variation trend of S. Specifically, Table 4 divides S into three groups, with G remaining consistent within each group, highlighting the variations in metrics across these groups. Figure 11b illustrates the variation trend of G. Here, Table 4 divides G into four groups, with S remaining unchanged within each group, demonstrating how the metrics change across the groups. Figure 11 visually demonstrates that CINet can maintain satisfactory results even with insufficient S and G samples, showcasing stronger robustness, particularly in SSIM, LPIPS, and StSc. Specifically, when G

Table 3 | Impact of reducing style samples on performance of the DPCDD Dataset.

S of seen inpainting character		PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSC ↑
S = 1315	CIDG[4]	18.1848	0.9260	0.0817	11.9105	0.8881
	DE-GAN[5]	17.8320	0.8221	0.1605	93.2365	0.7121
	CycleGAN[2]	13.8805	0.8070	0.1527	25.8557	0.6105
	RubGAN[3]	13.8123	0.8128	0.1466	29.7008	0.7520
	FD-Net[25]	18.2892	0.8401	0.1414	64.9951	0.8087
	RCRN[7]	19.5924	0.9297	0.0675	9.1496	0.9187
	Charformer[6]	16.0271	0.7506	0.1668	34.2922	0.6615
	Uformer[19]	17.9640	0.9087	0.0915	15.8036	0.8504
	Tformer[17]	18.4590	0.9192	0.0677	5.9465	0.9043
	CENet[29]	20.9787	0.8981	0.0881	37.7460	0.9023
	GSDM[14]	20.8786	0.9451	0.0491	3.5012	0.9455
	CINet(ours)	**21.5947**	**0.9493**	**0.0434**	**3.0026**	**0.9694**
S = 657	CIDG[4]	17.7657	0.9203	0.0928	16.6527	0.8842
	DE-GAN[5]	17.8188	0.7644	0.1651	95.8881	0.6941
	CycleGAN[2]	12.1792	0.7576	0.1810	38.0451	0.5767
	RubGAN[3]	14.0834	0.8071	0.1508	41.2603	0.7870
	FD-Net[25]	17.7891	0.8367	0.1466	68.0548	0.7782
	RCRN[7]	19.2927	0.9221	0.0734	10.7448	0.9169
	Charformer[6]	14.1731	0.5614	0.2690	110.9627	0.4956
	Uformer[19]	17.9784	0.9093	0.0930	17.7786	0.8448
	Tformer[17]	18.6731	0.9143	0.0660	5.3971	0.8995
	CENet[29]	20.6433	0.8925	0.0942	31.3222	0.8911
	GSDM[14]	20.4847	**0.9428**	0.0520	3.7486	0.9320
	CINet(ours)	**21.0008**	0.9282	**0.0482**	**3.7203**	**0.9638**
S = 329	CIDG[4]	17.4630	0.9020	0.1074	30.2657	0.8766
	DE-GAN[5]	17.5614	0.8181	0.1652	97.7782	0.7025
	CycleGAN[2]	11.3436	0.7212	0.2169	80.8598	0.5362
	RubGAN[3]	13.0118	0.7793	0.1752	43.9956	0.7089
	FD-Net[25]	16.3434	0.8015	0.1792	87.4964	0.6863
	RCRN[7]	18.2356	0.9037	0.0924	18.2225	0.8922
	Charformer[6]	14.4962	0.4556	0.2628	121.8568	0.6082
	Uformer[19]	17.6506	0.8984	0.0973	21.1528	0.8320
	Tformer[17]	18.0993	0.9106	0.0750	9.8653	0.9025
	CENet[29]	20.0241	0.8293	0.1087	39.5338	0.8554
	GSDM[14]	20.2234	**0.9411**	0.0548	4.2930	0.9281
	CINet(ours)	**20.3057**	0.9244	**0.0538**	**4.2230**	**0.9483**

The optimal results are shown in bold, and the sub-optimal results are underlined. S denotes style samples.

changes (from 5 to 1), SSIM, LPIPS, and StSc vary smoothly. PSNR decreases slightly (with a maximum change of 1.1614 dB), while FID increases slightly (with a maximum change of 1.8070). When S changes (from 2629 to 329), SSIM, LPIPS, and StSc show better robustness (with maximum changes of 0.0349, 0.0233, and 0.0275, respectively). PSNR decreases moderately (with a maximum change of 2.6331 dB), and FID increases (with a maximum change of 3.2777).

Table 5 reports the influence of degradation levels on the effectiveness of various inpainting methods. For slight degradation (10–20%), the loss of image information is relatively small, allowing most methods to effectively leverage the remaining features, thus achieving better performance. However, under severe degradation (30–40%), significant loss of image information results in poor performance for methods that rely solely on image features. In comparison, our method consistently produces higher-quality results across all degradation levels. Notably, under conditions of severe degradation, CINet compensates for the loss of image information by utilizing glyph structure information, ensuring superior robustness. Specifically, under slight degradation conditions, most methods show good performance in terms of PSNR and SSIM; however, CINet achieves higher PSNR and SSIM values, with notable improvements in LPIPS and FID. For example, under 20% degradation on the DHWD dataset, CINet attains a PSNR of 28.5654 dB and an StSc of 0.8705, outperforming all other methods. CINet also achieves an FID score of 6.9545, which is 2.4368 lower than the second-best method, GSDM. Under high degradation conditions, the performance of most methods (e.g., RubGAN and FD-Net) significantly declines, while CINet maintains relatively stable performance. Moreover, CINet shows notable advantages in LPIPS and FID. For example, on the DPCCD dataset with 40% degradation, CINet achieves an FID of 6.3836,

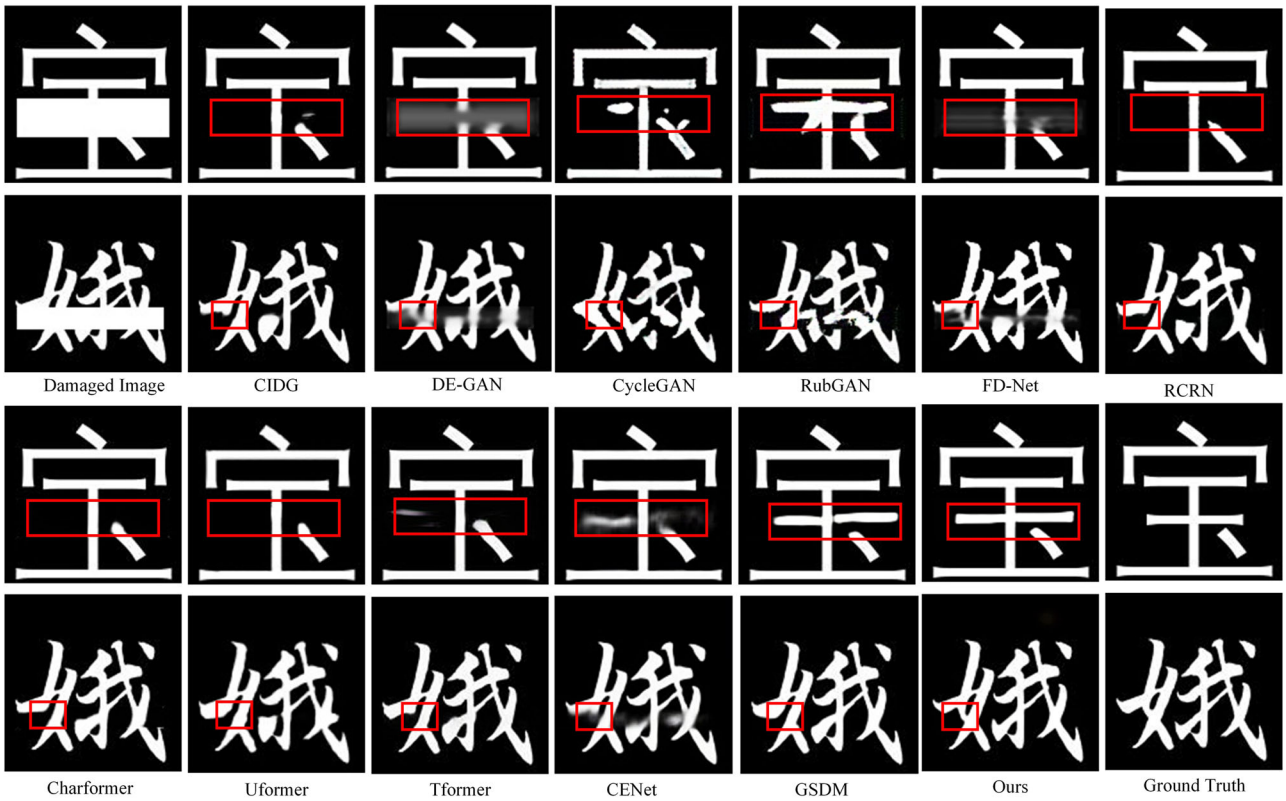

Fig. 10 | Restoration results of different methods on damaged DPCCD images. The red boxes highlight regions for the comparison of local inpainting results across methods.

Table 4 | Impact of reducing glyph instances and style samples on CINet performance.

G of seen inpainting character	S of seen inpainting character	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSc ↑
5	2629	22.5642	0.9501	0.0365	2.5185	0.9728
	1315	21.5947	0.9493	0.0434	3.0026	0.9694
	657	21.0008	0.9282	0.0482	3.7203	0.9638
	329	20.3057	0.9244	0.0538	4.2230	0.9483
3	2629	22.1492	0.9525	0.0387	2.6915	0.9680
	1315	21.1227	0.9474	0.0461	3.3452	0.9634
	657	20.3279	0.9416	0.0533	4.2516	0.9567
	329	19.5161	0.9350	0.0610	5.5980	0.9405
1	2629	21.7690	0.9504	0.0413	2.7523	0.9677
	1315	20.4333	0.9418	0.0511	4.1197	0.9568
	657	20.2268	0.9414	0.0547	4.5300	0.9535
	329	19.3160	0.9155	0.0646	6.0300	0.9437

significantly outperforming the second-best method, Tformer (15.2614). Additionally, its LPIPS and StSc scores are 0.0820 and 0.9014, respectively, demonstrating clear superiority over other methods. To summarize, CINet delivers excellent performance across different degradation levels (10–40%) on the DPCCD and DHWD datasets, showcasing remarkable adaptability to complex scenes. Figure 12 shows the restoration results of the DPCCD and DHWD datasets. Our method restores the glyph structure more accurately, while other methods exhibit distortion and artifacts. Furthermore, we use Grad-CAM for visualization (as shown in Fig. 13). We overlay the heat maps generated by different methods onto the damaged images to highlight the areas the model focuses on. Red areas indicate high attention, while blue represents low attention. In CINet, the red-highlighted areas are concentrated on the glyph structure or missing parts, demonstrating that the model correctly focuses on the restoration areas, thereby achieving better results. In contrast, other models (e.g., Tformer) fail to adequately focus on the glyph structure, or their highlighted areas do not concentrate on glyph-related regions (e.g., CIDG, RCRN), leading to glyph distortion.

In Table 6, we report the performance of various methods on the DID dataset. Our method outperforms others across most metrics, with the exception of FID, where it ranks second, demonstrating significant advantages in real-inscription image restoration. This makes our model more suitable for real-world scenarios. Specifically, although CINet's FID is 16.2541, which is only 0.4457 higher than the best-performing Tformer (15.8084), it surpasses both GSDM and Tformer in PSNR, SSIM, LPIPS, and StSc. CINet achieves SSIM and StSc values of 0.9564 and 0.9534, respectively, outperforming Tformer by 0.0115 and 0.0305. It also outperforms the second-best method, GSDM, by 0.9711 dB in PSNR. Furthermore, CINet has the lowest LPIPS score of 0.0370, which is significantly better than other methods, such as CIDG (LPIPS = 0.0628) and Charformer (LPIPS = 0.2935). Figure 14 displays the restoration results, where CINet ensures consistent restoration of the glyph structure, while comparison models (e.g., Tformer) retain unnecessary strokes. GSDM performs well in handling slight degradations but is less effective at removing spurious strokes, often resulting in the retention of incorrect structural information. Figure 15 presents the Grad-CAM visualization results, showcasing how CINet effectively focuses on the glyph structure, distinguishing real from fake features. In contrast, other models (e.g., Tformer and CIDG) fail to concentrate on the glyph regions accurately and instead focus on irrelevant areas, leaving meaningless strokes in the restored images.

Ablation study

To evaluate the impact of different components on restoration performance, we analyze the performance fluctuations after removing each module on the DID dataset, as shown in Table 7. The results indicate that the removal of

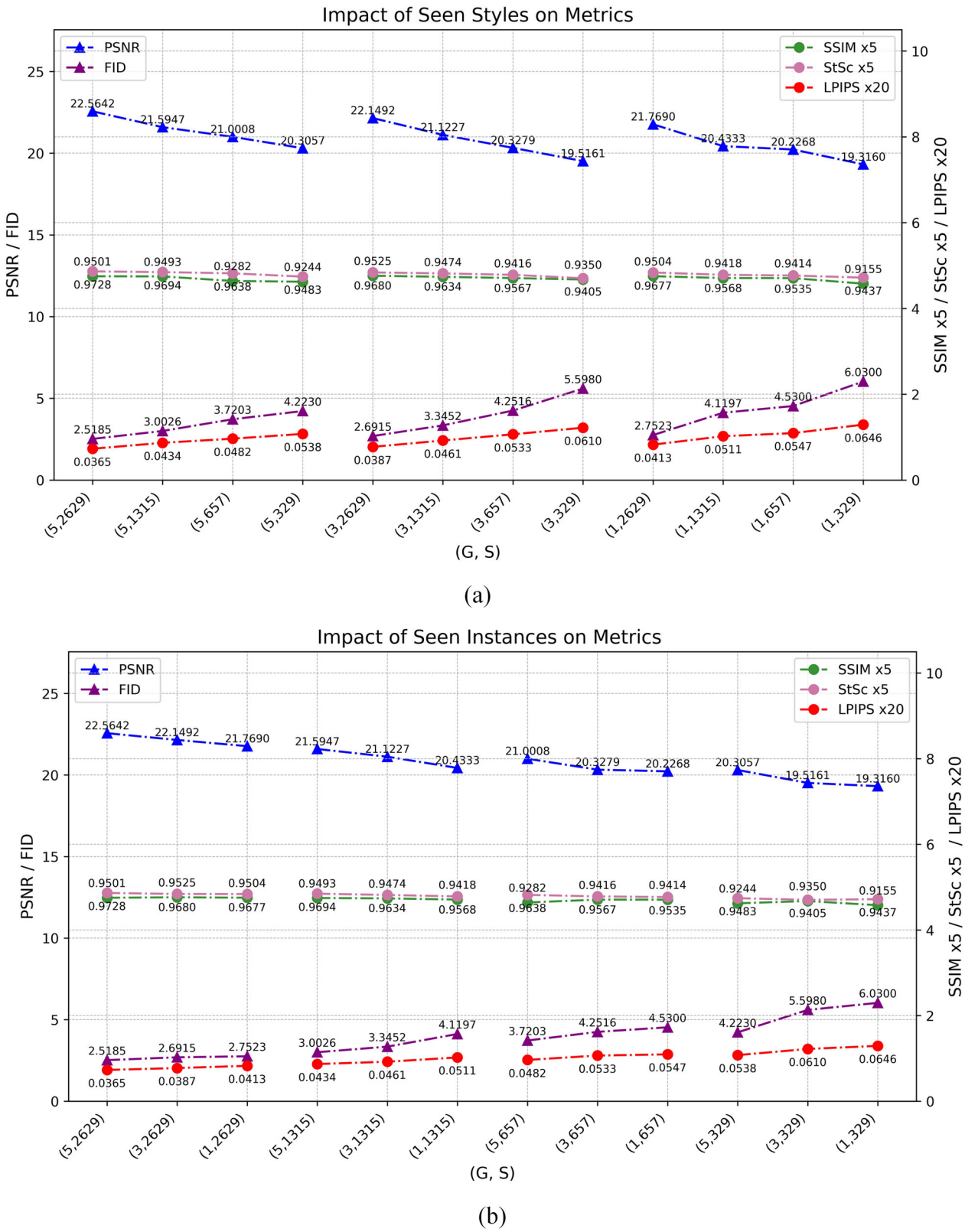

Fig. 11 | Impact of G and S samples on CINet performance. a Impact of seen style samples (S) on performance. **b** Impact of seen glyph instance samples(G) on performance.

any module negatively affects CINet's performance, underscoring the importance of module cooperation in achieving high performance. When removing a single component, the performance degradation is relatively minor, with significant fluctuations occurring only in specific metrics. For example, removing the FSM module leads to more noticeable increases in LPIPS and FID, which rise by 0.0018 and 3.5012, respectively, while its effect on SSIM is negligible. In contrast, removing the S_{emb} has a more substantial impact on StSc, reducing it from 0.9534 to 0.9301. When multiple modules are removed together, the performance degradation is much more pronounced, highlighting strong synergistic effects among the modules. For

Fig. 12 | Restoration results of different methods on DPCCD/DHWD. a Visualization of different methods on DPCCD. **b** Visualization of different methods on DHWD.

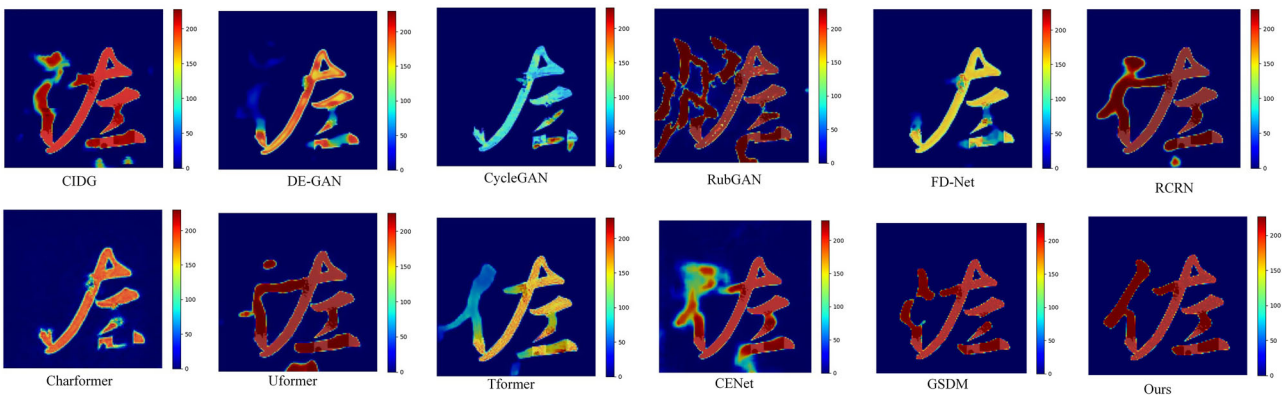

Fig. 13 | Visualization of Grad-CAM results on DPCCD. Red areas indicate high attention, and blue areas indicate low attention.

example, in the CINet-FSM-CA_{TEX}-CA_{IMG}-S_{emb}-E_{TEX} or CINet-FSM-CA_{TEX}-CA_{IMG}-S_{emb} configurations, all metrics show a significant decline, and the magnitude of this degradation is far greater than when removing a single module. Specifically, when all modules are removed, PSNR decreases by 1.2043 dB, and FID increases by 21.2852. It is noteworthy that even when all components are removed, the CSB branch, relying solely on CA_{IMG_TEX} to transfer glyph information to the IB branch, still performs reasonably well, achieving a PSNR of 20.5643 dB and an FID of 37.5393. These results outperform most of the comparison methods (as shown in Table 6), highlighting the critical role of glyph information in inpainting inscription images.

To evaluate the network's adaptability and restoration performance under different damage shapes, we simulate circular, rectangular, and square damage patterns at a 20% degradation level on the DHWD and DPCCD datasets, with results shown in Table 8. The results are visualized in Fig. 16. The findings indicate that CINet exhibits robust restoration capabilities, showing minimal sensitivity to different damage shapes. This reflects the model's strong generalizability across various damage patterns. Specifically, on the DHWD dataset, our model demonstrates consistent performance with only minor fluctuations across all metrics. The maximum variations are 0.0027 for SSIM, 0.0057 for LPIPS, and 0.0052 for StSc. On the DPCCD dataset, CINet similarly shows excellent robustness to different damage shapes. Notably, the maximum fluctuations are 0.0044 for SSIM (ranging from 0.9501 to 0.9545), 0.0027 for LPIPS (from 0.0365 to 0.0392), 0.4527 for FID (from 2.4913 to 2.9440), and 0.0059 for StSc (from 0.9728 to 0.9787).

More visualization results

To evaluate the generalization capability of the proposed method across different Chinese calligraphy styles, we perform restoration experiments

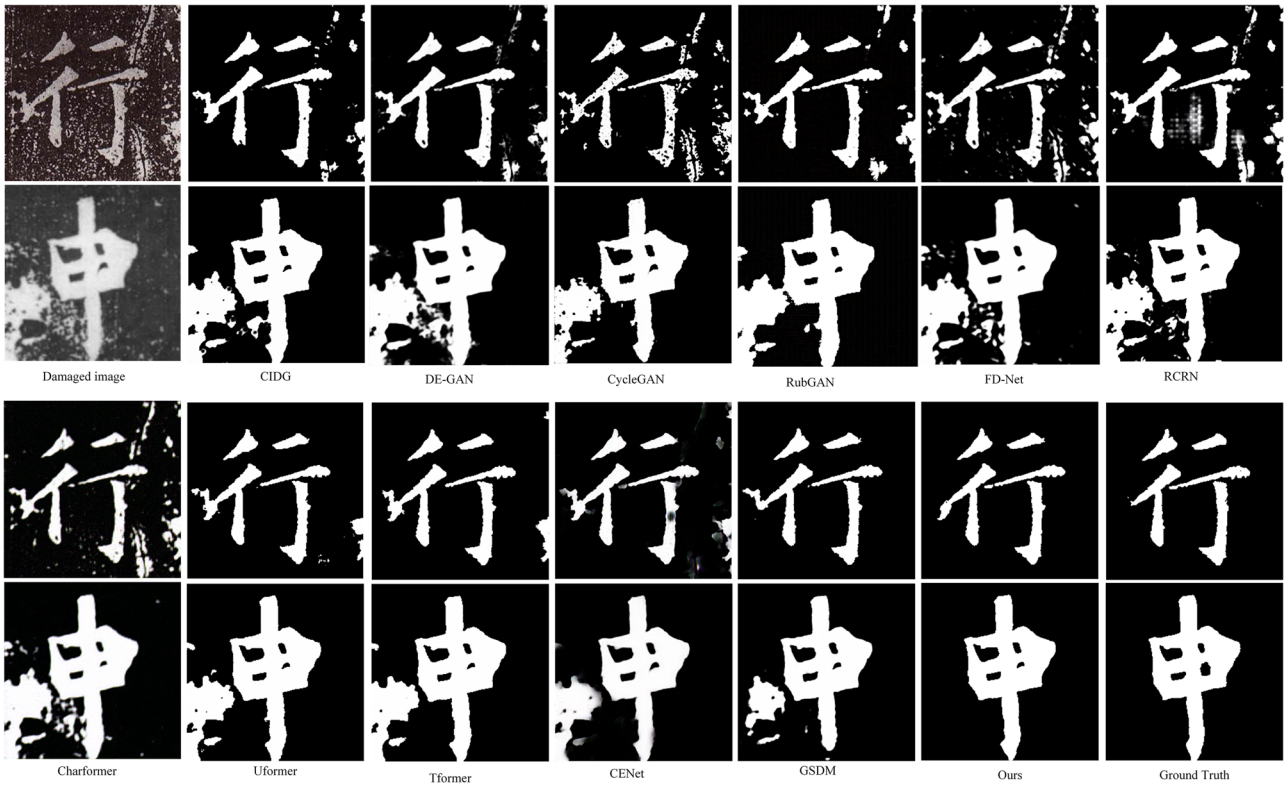

Fig. 14 | Inpainting results of different methods on DID. Each row shows, from left to right: Damaged image, CIDG, DE-GAN, CycleGAN, RubGAN, FD-Net, RCRN, Charformer, Uformer, Tformer, CENet, GSDM, Ours and Ground Truth.

Table 5 | Comparison of different methods under varying degradation ratios (10–40%) on damaged DPCCD and DHWD datasets.

DataSet		DPCCD				DHWD			
Damage Ratio		**10%**	**20%**	**30%**	**40%**	**10%**	**20%**	**30%**	**40%**
PSNR ↑	CIDG[4]	23.8485	17.9637	15.6344	14.3110	32.5975	27.0959	23.6979	21.1915
	DE-GAN[5]	21.5729	17.4547	14.0916	12.4792	22.9489	19.0931	19.5765	16.1321
	CycleGAN[2]	16.3694	15.0276	12.1797	11.4907	12.7699	12.5543	12.0983	8.7998
	RubGAN[3]	16.0821	12.3080	10.6780	8.6255	23.6705	15.6063	14.6452	16.1002
	FD-Net[25]	22.7529	17.5490	13.8879	12.5476	27.6416	20.4685	20.2979	16.8503
	RCRN[7]	27.1793	20.0975	17.9561	16.3993	25.7484	21.5700	20.9555	20.3070
	Charformer[6]	16.9728	13.9899	12.1706	11.1936	22.4961	18.9827	17.0869	16.0577
	Uformer[19]	22.9206	18.1223	16.3102	13.8839	28.5608	26.2205	23.2619	20.9709
	Tformer[17]	24.6870	18.5650	17.1622	15.3204	31.9444	27.9480	25.1058	22.4783
	CENet[29]	26.9407	21.9001	18.1533	16.6877	32.0849	21.5519	18.0554	16.9368
	GSDM[14]	**27.3847**	**22.2099**	18.9256	16.7105	31.3204	26.5663	25.1741	**23.2469**
	CINet(ours)	27.0752	21.9040	**19.0391**	**16.9545**	**33.6503**	**28.5654**	**25.3990**	22.9516
SSIM ↑	CIDG[4]	0.9708	0.9235	0.8879	0.8198	0.9850	0.9563	0.9149	0.8585
	DE-GAN[5]	0.5544	0.5088	0.4621	0.4013	0.9278	0.8537	0.7955	0.7236
	CycleGAN[2]	0.8763	0.8457	0.7811	0.7586	0.7351	0.6979	0.4804	0.4396
	RubGAN[3]	0.8735	0.7581	0.6889	0.5699	0.9192	0.7061	0.6686	0.6972
	FD-Net[25]	0.9170	0.6117	0.4419	0.4107	0.9594	0.8551	0.8157	0.6964
	RCRN[7]	0.9737	0.9409	0.9086	0.8795	0.9770	0.9425	0.9226	0.8992
	Charformer[6]	0.8724	0.7155	0.7090	0.4127	0.9000	0.8268	0.6558	0.7311
	Uformer[19]	0.9529	0.9097	0.8857	0.8134	0.9562	0.9466	0.9102	0.8515
	Tformer[17]	0.9713	0.9281	0.8859	0.8519	0.9871	**0.9692**	0.9421	0.9059
	CENet[29]	0.9495	0.8828	0.8373	0.7234	0.9796	0.9119	0.8459	0.6361
	GSDM[14]	0.9684	0.9402	**0.9161**	0.8455	0.9811	0.9556	0.9389	0.9058
	CINet(ours)	**0.9788**	**0.9573**	0.9142	**0.8981**	**0.9880**	0.9689	**0.9423**	**0.9102**
LPIPS ↓	CIDG[4]	0.0367	0.0897	0.1260	0.1766	0.0224	0.0576	0.0971	0.1448

Table 5 (continued) | Comparison of different methods under varying degradation ratios (10–40%) on damaged DPCCD and DHWD datasets.

DataSet		DPCCD				DHWD			
	DE-GAN[5]	0.1437	0.2223	0.3295	0.3940	0.1127	0.1941	0.2390	0.3010
	CycleGAN[2]	0.1082	0.1273	0.1903	0.2079	0.2441	0.2608	0.3755	0.4007
	RubGAN[3]	0.1211	0.2101	0.2566	0.3338	0.1342	0.3186	0.3171	0.2953
	FD-Net[25]	0.0778	0.1996	0.3397	0.3966	0.0587	0.1782	0.2120	0.2979
	RCRN[7]	0.0250	0.0616	0.0926	0.1177	0.0424	0.0901	0.1063	0.1320
	Charformer[6]	0.1199	0.1937	0.2366	0.3058	0.1491	0.2436	0.3602	0.3049
	Uformer[19]	0.0489	0.0909	0.1143	0.1777	0.0637	0.0722	0.1071	0.1511
	Tformer[17]	0.0261	0.0643	0.0858	0.1203	**0.0179**	**0.0368**	**0.0611**	**0.0914**
	CENet[29]	0.0403	0.0823	0.1268	0.1975	0.0328	0.1182	0.1842	0.3268
	GSDM[14]	0.0281	0.0543	0.0760	0.1219	0.0297	0.0583	0.0810	0.1100
	CINet(ours)	**0.0186**	**0.0375**	**0.0590**	**0.0820**	0.0200	0.0441	0.0723	0.1029
FID ↓	CIDG[4]	4.5217	21.9642	34.9379	69.1312	3.3452	15.5087	33.5072	61.3743
	DE-GAN[5]	82.9050	106.0662	139.7068	176.1016	84.9855	103.6033	175.3730	145.0369
	CycleGAN[2]	23.9995	26.4705	45.1659	41.9275	78.9258	70.8618	173.5766	157.7194
	RubGAN[3]	30.1559	61.1383	81.5016	92.0147	128.6956	206.5620	255.3818	185.5508
	FD-Net[25]	30.2129	91.9941	162.6583	203.0982	26.0509	89.4191	155.0165	124.9439
	RCRN[7]	6.1782	9.7630	25.8691	38.6952	5.2834	11.5529	18.6234	30.3388
	Charformer[6]	49.3760	76.0000	100.5720	148.0570	211.4300	257.8482	286.1490	281.6109
	Uformer[19]	10.3924	16.8941	21.4015	54.7658	30.4457	34.1864	56.2771	74.2093
	Tformer[17]	2.1062	5.5788	8.6455	15.2614	5.8980	12.1144	30.1857	47.7919
	CENet[29]	16.7125	31.2776	69.0545	114.4214	13.7978	77.7357	71.3762	246.1889
	GSDM[14]	10.9715	12.9593	13.1736	23.9740	12.4396	9.3913	24.4904	53.6999
	CINet(ours)	**1.5773**	**2.5201**	**4.9128**	**6.3836**	**2.4655**	**6.9545**	**14.0550**	**23.4199**
StSc ↑	CIDG[4]	0.9803	0.9266	0.8609	0.7492	**0.9753**	0.8271	0.6994	0.5699
	DE-GAN[5]	0.9471	0.6147	0.3757	0.2718	0.2187	0.0919	0.1299	0.0889
	CycleGAN[2]	0.7652	0.7020	0.6268	0.5583	0.1383	0.1329	0.1284	0.1125
	RubGAN[3]	0.7822	0.6957	0.5368	0.3265	0.7899	0.4962	0.5573	0.3581
	FD-Net[25]	0.9524	0.8131	0.3726	0.2606	0.4633	0.2030	0.2132	0.0980
	RCRN[7]	0.9876	0.9362	0.8686	0.8226	0.4025	0.2874	0.2994	0.2583
	Charformer[6]	0.8821	0.7877	0.6966	0.6332	0.5176	0.4686	0.1700	0.2569
	Uformer[19]	0.9268	0.8323	0.8266	0.6666	0.5441	0.4140	0.3812	0.3108
	Tformer[17]	0.9812	0.8948	0.8686	0.7867	0.8789	0.7768	0.6754	0.4973
	CENet[29]	0.9838	0.9082	0.8259	0.6297	0.9162	0.6541	0.4912	0.1362
	GSDM[14]	0.9883	0.9512	0.8948	0.8220	0.9636	0.8225	0.6748	0.4575
	CINet(ours)	**0.9902**	**0.9684**	**0.9385**	**0.9014**	0.9679	**0.8705**	**0.7306**	**0.6023**

The optimal results are shown in bold, and the sub-optimal results are underlined.

Table 6 | Comparison of performance on real-inscription images.

DID	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSc ↑
CIDG[4]	19.7501	0.9407	0.0628	25.5439	0.9098
DE-GAN[5]	19.9100	0.9076	0.1430	122.9757	0.8952
CycleGAN[2]	16.9535	0.8826	0.1056	43.6596	0.6900
RubGAN[3]	17.6579	0.5975	0.1841	115.0252	0.8180
FD-Net[25]	18.9690	0.8811	0.1743	147.9551	0.8355
RCRN[7]	18.7303	0.9123	0.1003	78.1304	0.8559
Charformer[6]	18.1647	0.4742	0.2935	204.4901	0.7948
Uformer[19]	18.5368	0.9312	0.0585	18.5603	0.8355
Tformer[17]	20.0371	0.9449	0.0475	**15.8084**	0.9229
CENet[29]	20.4213	0.8961	0.1285	92.8850	0.9112
GSDM[14]	20.7975	0.9347	0.0917	81.6815	0.9199
CINet	**21.7686**	**0.9564**	**0.0370**	16.2541	**0.9534**

The optimal results are shown in bold, and the sub-optimal results are underlined.

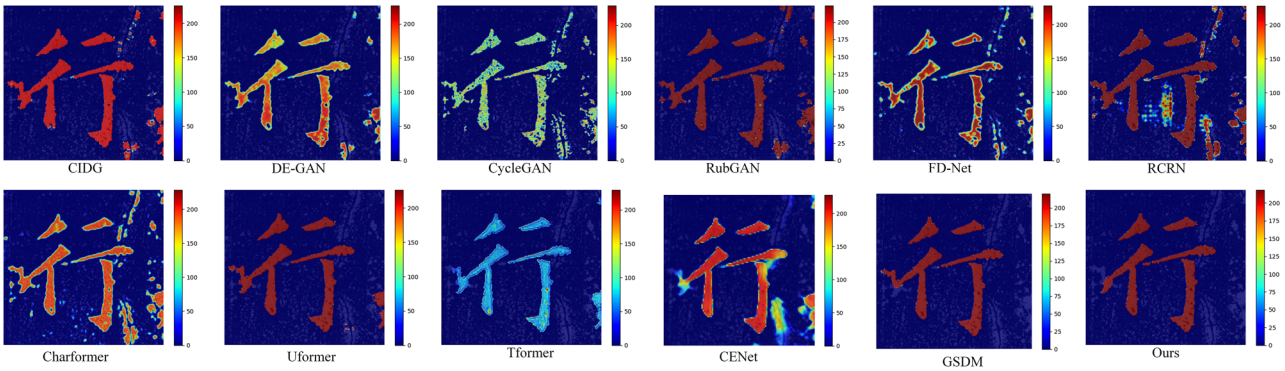

Fig. 15 | Grad-CAM visualization results on DID. Red areas indicate high attention, while blue areas indicate low attention.

Table 7 | Ablation experiments of different components.

Methods	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSc ↑
ClNet	**21.7686**	**0.9564**	**0.0370**	**16.2541**	**0.9534**
ClNet-FSM	21.6633	0.9551	0.0388	19.7553	0.9476
ClNet-S_{emb}	21.3994	0.9534	0.0382	16.6070	0.9301
ClNet-E_{TEX}	21.5526	0.9551	0.0381	18.8814	0.9505
ClNet-CA_{IMG}	21.7001	0.9550	0.0373	16.2934	0.9491
ClNet-CA_{TEX}	21.4337	0.9438	0.0389	16.4120	0.9389
ClNet-CA_{TEX}-CA_{IMG}	21.3917	0.9533	0.0382	16.5173	0.9345
ClNet-FSM- CA_{TEX}-CA_{IMG}	21.6970	0.9483	0.0387	26.5635	0.9461
ClNet-FSM- CA_{TEX}-CA_{IMG}-S_{emb}	21.5443	0.9216	0.0386	32.5536	0.9389
ClNet-FSM- CA_{TEX}-CA_{IMG}-S_{emb}-E_{TEX}	20.5643	0.9335	0.0431	37.5393	0.9127

The optimal results are shown in bold, and the sub-optimal results are underlined.

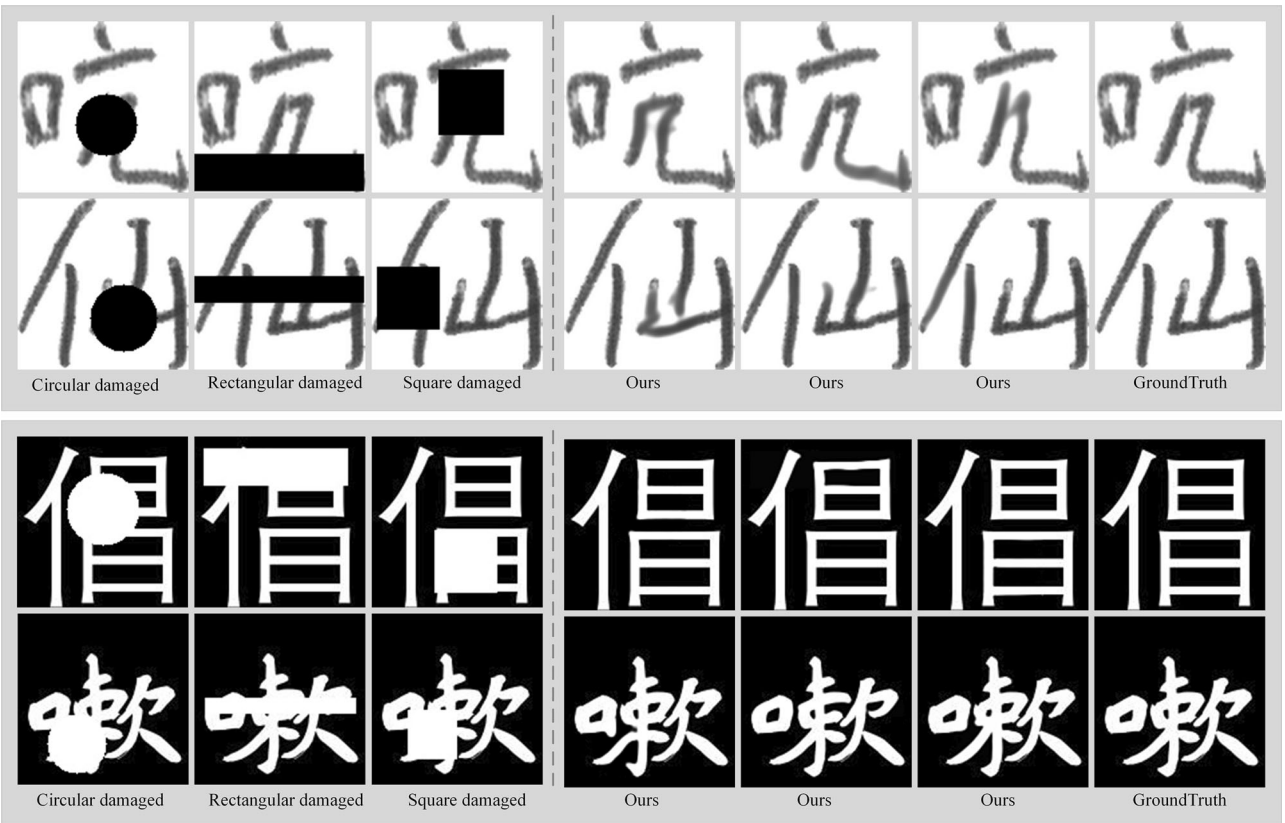

Fig. 16 | Restoration results of our method on DHWD/DPCCD for different damage shapes. The left side of the dashed line shows examples of different damage shapes, and the right side shows the restoration results of our method.

Fig. 17 | Visualization of restoration results for Yan, Ou, and Liu scripts. The first row shows the damaged images, and the second row shows the results obtained using our proposed method. **a** Visualization of the testing results for Liu scripts. **b** Visualization of the testing results for Ou scripts. **c** Visualization of the testing results for Yan scripts.

during the testing stage on Yan Zhenqing's Epitaph of Guo Xuji (Yan script), Ouyang Xun's Jiuchenggong Liquan Inscription (Ou script), and Liu Gongquan's Diamond Sutra (Liu script) (see Fig. 17). The DID training set contains only Yan script samples, with no Ou or Liu script samples. Moreover, for the Yan script, the training and testing sets are strictly non-overlapping. The visualization results demonstrate that CINet can accurately inpaint glyphs and maintain style consistency in both unseen fonts (Ou and Liu scripts) and seen fonts with different samples (Yan script), evidencing its effectiveness and robustness under varying font style scenarios.

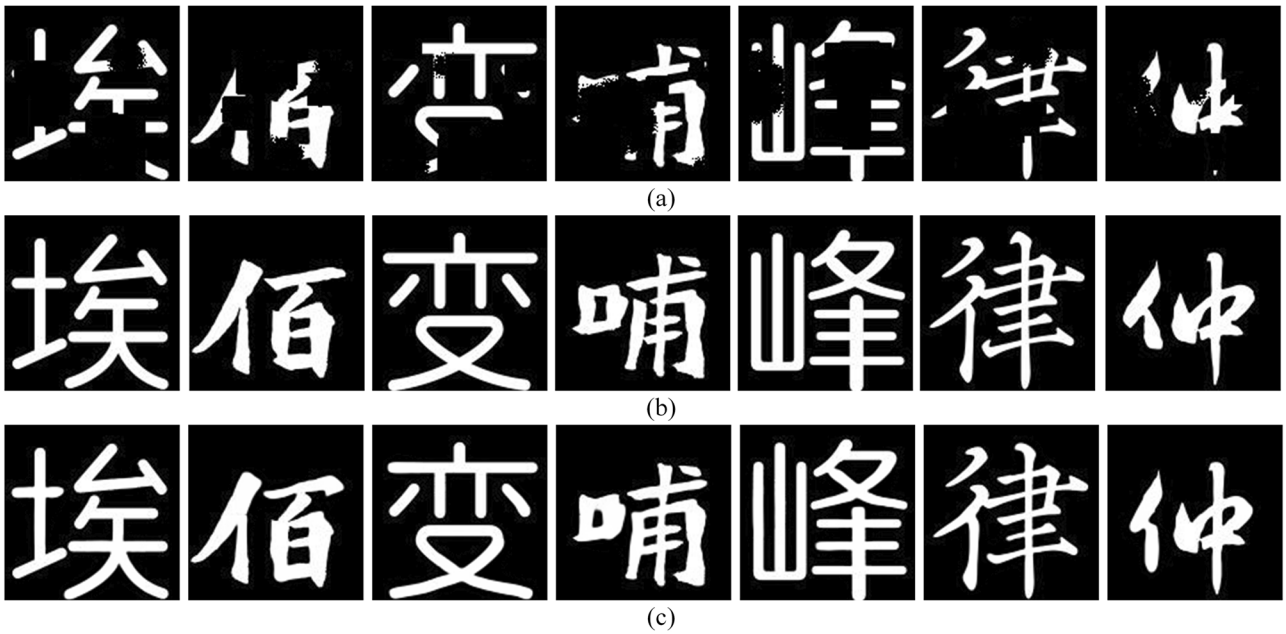

Fig. 18 | Visualization of successful restoration examples. a DPCCD image with 40% degradation ratio. **b** Ground-truth images. **c** Inpainting results of our method.

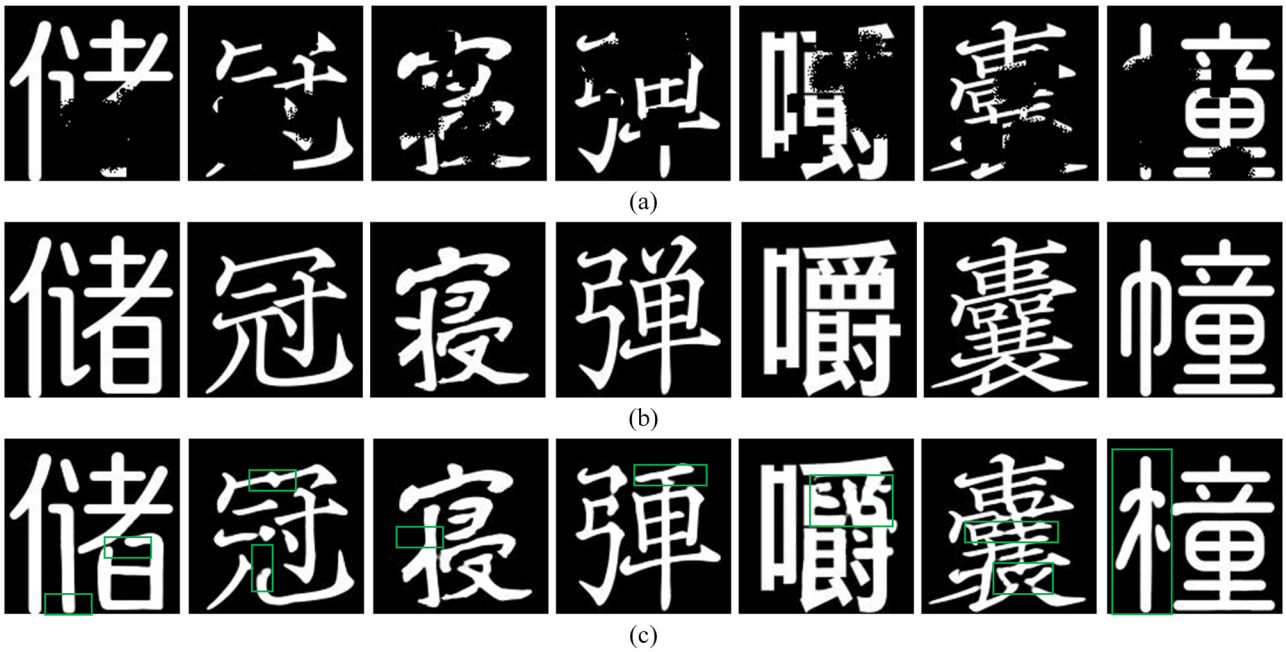

Fig. 19 | Visualization of failed restoration examples. a DPCCD images with 40% degradation ratio. **b** Ground-truth images. **c** Inpainting results of our method.

To further validate the advantages and limitations of the proposed method, we select a 40% degradation ratio from the DPCCD dataset as the testing scenario, which provides a challenging setting and effectively evaluates the model's inference ability under conditions of missing information. As shown in Table 5, under the 40% damage condition, CINet still achieves the best performance in all indicators. However, the visualization results (see Figs. 18, 19) show that some characters are still not fully restored. Specifically, samples with successful inpainting (see Fig. 18) generally retain most of their crucial structural elements, enabling the model to accurately infer missing parts based on existing structural information. In contrast, failed inpainting cases (see Fig. 19) often exhibit substantial loss of core strokes and numerous strokes or overlapping components, which significantly increase the restoration difficulty and lead to missing strokes, structural distortions, or incorrect completions, as highlighted by the green boxes. This indicates that while the proposed method performs well overall, it still faces challenges and has potential for improvement in inpainting Chinese characters with severe key structure loss and multiple overlapping components.

Discussion

We propose CINet, a specialized image inpainting network built with an in-depth understanding of Chinese character structures. CINet adopts a dual-branch architecture. The first branch, the CSB, generates high-quality representations of Chinese characters. Specifically, we utilize the text encoder from the CLIP model, pretrained for Chinese character image recognition, to provide additional supervisory information to the CSB. The second branch, the IB, incorporates a cross-attention mechanism to inject

Table 8 | Impact of different mask shapes on CINet performance

Mask Shape	DHWD					DPCCD				
	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSc ↑	PSNR ↑	SSIM ↑	LPIPS ↓	FID ↓	StSc ↑
Circle	24.0742	0.9356	0.0637	10.0686	0.8097	21.3738	0.9526	0.0392	2.9440	0.9787
Rectangle	24.6600	0.9383	0.0675	8.2798	0.8067	22.5642	0.9501	0.0365	2.5185	0.9728
Square	24.4833	0.9376	0.0618	9.3718	0.8119	21.8370	0.9545	0.0374	2.4913	0.9739

key glyph information, guiding the inpainting task. This enables the model to be more sensitive to glyph features and compensates for the limitations of relying solely on degraded image feature extraction. The cross-modal design of CINet also effectively addresses the challenge of insufficient inscription image data. Additionally, to improve the network's ability to capture and preserve style, we integrate a style embedding module. This enhances the model's precision in maintaining style consistency during inpainting. We demonstrate the superiority of CINet across multiple datasets, especially in scenarios with limited data and complex degradation, showing greater robustness and adaptability.

Data availability

The DID, DHWD, and DPCCD datasets are available via the following link: https://github.com/liuyunjing0306/CINet.

Code availability

The code for this study is available via the following link. https://github.com/liuyunjing0306/CINet.

Received: 8 June 2025; Accepted: 21 September 2025;
Published online: 02 October 2025

References

1. Goodfellow, I. et al. Generative adversarial networks. *Commun. ACM* **63**, 139–144 (2020).
2. Zhu, J.-Y., Park, T., Isola, P. & Efros, A. A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *IEEE Int. Conf. Comput. Vision*, 2242–2251 (IEEE, 2017).
3. Sun, G., Zheng, Z. & Zhang, M. End-to-end rubbing restoration using generative adversarial networks. Preprint at https://doi.org/10.48550/arXiv.2205.03743 (2022).
4. Zhang, J., Guo, M. & Fan, J. A novel generative adversarial net for calligraphic tablet images denoising. *Multimedia Tools Appl.* **79**, 119–140 (2020).
5. Souibgui, M. A. & Kessentini, Y. De-gan: a conditional generative adversarial network for document enhancement. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 1180–1191 (2022).
6. Shi, D. et al. Charformer: a glyph fusion based attentive framework for high-precision character image denoising. In *Proc. 30th ACM International Conference on Multimedia*. 1147–1155 (Association for Computing Machinery, 2022).
7. Shi, D. et al. Rcrn: real-world character image restoration network via skeleton extraction. In *Proc. 30th ACM international conference on multimedia*. 1177–1185 (Association for Computing Machinery, 2022).
8. Radford, A. et al. Learning transferable visual models from natural language supervision. *Int. Conf. Mach. Learn.* **139**, 8748–8763 (2021).
9. Yu, H., Wang, X., Li, B. & Xue, X. Chinese text recognition with a pre-trained clip-like model through image-ids aligning. In *Proc. IEEE/CVF International Conference on Computer Vision* 11909–11918 (IEEE, 2023).
10. Duan, C., Fu, P., Guo, S., Jiang, Q. & Wei, X. Odm: a text-image further alignment pre-training approach for scene text detection and spotting. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15587–15597 (IEEE, 2024).
11. Zhao, L., Guo, Q., Li, X. & Wang, S. Clii: Visual-text inpainting via cross-modal predictive interaction. Preprint at https://doi.org/10.48550/arXiv.2407.16204 (2024).
12. Li, J., Wang, N., Zhang, L., Du, B. & Tao, D. Recurrent feature reasoning for image inpainting. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 7757–7765 (IEEE, 2020).
13. Lugmayr, A. et al. Repaint: Inpainting using denoising diffusion probabilistic models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 11451–11461 (IEEE, 2022).
14. Zhu, S. et al. Text image inpainting via global structure-guided diffusion models. *AAAI Conf. Art. Intell.* **38**, 7775–7783 (2024).
15. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst*. **30**, 5999–6009 (2017).
16. Li, W. et al. Mat: Mask-aware transformer for large hole image inpainting. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 10748–10758 (IEEE, 2022).
17. Deng, Y., Hui, S., Zhou, S., Meng, D. & Wang, J. T-former: an efficient transformer for image inpainting. In *Proc. 30th ACM International Conference on Multimedia.* 6559-6568 (Association for Computing Machinery, 2022).
18. Chen, S., Atapour-Abarghouei, A., Zhang, H. & Shum, H. P. H. Mxt: Mamba x transformer for image inpainting. Preprint at https://doi.org/10.48550/arXiv.2407.16126 (2024).
19. Wang, Z. et al. Uformer: a general u-shaped transformer for image restoration. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 17662–17672 (IEEE, 2022).
20. Huang, W. et al. Sparse self-attention transformer for image inpainting. *Pattern Recognit*. **145**, 109897 (2024).
21. Chen, Y., Xia, R., Yang, K. & Zou, K. Gcam: lightweight image inpainting via group convolution and attention mechanism. *Int. J. Mach. Learn. Cybern.* **15**, 1815–1825 (2024).
22. Wang, H.-H., Tsai, F.-J., Lin, Y.-Y. & Lin, C.-W. Tanet: triplet attention network for all-in-one adverse weather image restoration. *Asian Conf. Comput. Vision* **15475 LNCS**, 3–19 (2025).
23. Chen, Y., Xia, R., Yang, K. & Zou, K. Dnnam: image inpainting algorithm via deep neural networks and attention mechanism. *Appl. Soft Comput*. **154**, 111392 (2024).
24. Xing, C. & Ren, Z. Binary inscription character inpainting based on improved context encoders. *IEEE Access* **11**, 55834–55843 (2023).
25. Xiong, W., Yue, L., Zhou, L., Wei, L. & Li, M. Fd-net: a fully dilated convolutional network for historical document image binarization. *Pattern Recognit. Comput. Vis.* **13019 LNCS**, 518–529 (2021).
26. Li, H. et al. Generative character inpainting guided by structural information. *Visual Comput.* **37**, 2895–2906 (2021).
27. Song, G., Li, J. & Wang, Z. Occluded offline handwritten chinese character inpainting via generative adversarial network and self-attention mechanism. *Neurocomputing* **415**, 146–156 (2020).
28. Kingma, D. P. & Welling, M. Auto-encoding variational bayes. In *Int. Conf. Learn. Represent.* (International Conference on Learning Representations, 2014).
29. Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T. & Efros, A. A. Context encoders: feature learning by inpainting. In *Proc IEEE Conference on Computer Vision and Pattern Recognition*. 2536–2544 (IEEE, 2016).

30. Zhao, L., Yuan, Z. & Lou, Y. Cross auto-encoder for inscription character inpainting. In *Int. Joint Conference on Neural Networks.* 1–8 (IEEE, 2024).
31. Lugo-Torres, G., Peralta-Rodriguez, D. A., Valdez-Rodriguez, J. E. & Calvo, H. Enhancing document digitization: image denoising with a cycle generative adversarial network. In *IEEE Symp. Ser. Comput. Intell*. 1461–1466 (IEEE, 2023).
32. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. 770–778 (IEEE, 2016).
33. Zhang, K., Zuo, W., Chen, Y., Meng, D. & Zhang, L. Beyond a gaussian denoiser: residual learning of deep cnn for image denoising. *IEEE Trans. Image Process.* **26**, 3142–3155 (2017).
34. Gurrola-Ramos, J., Dalmau, O. & Alarcon, T. E. A residual dense u-net neural network for image denoising. *IEEE Access* **9**, 31742–31754 (2021).
35. Dai, G. et al. Disentangling writer and character styles for handwriting generation. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* 5977–5986 (IEEE, 2023).
36. Zhang, Y., Zhang, Y. & Cai, W. Separating style and content for generalized style transfer. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition.* 8447–8455 (IEEE, 2018).
37. Kim, H. & Sohn, K.-A. How positive are you: text style transfer using adaptive style embedding. In *Int. Conf. Comput. Linguist*. 2115–2125 (Association for Computational Linguistics, 2020).
38. Mehta, S., Rastegari, M., Shapiro, L. & Hajishirzi, H. Espnetv2: A light-weight, power-efficient, and general-purpose convolutional neural network. In *IEEE Comput. Soc. Conf. Comput. Vision Pattern Recognit*. 9182–9192 (IEEE, 2019).
39. Liu, C.-L., Yin, F., Wang, D.-H. & Wang, Q.-F. Online and offline handwritten chinese character recognition: Benchmarking on new databases. *Pattern Recognit.* **46**, 155–162 (2013).
40. Tang, S. & Lian, Z. Write like you: Synthesizing your cursive online chinese handwriting via metric-based meta learning. *Comput. Graphics Forum* **40**, 141–151 (2021).
41. Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. In *Int. Conf. Learn. Represent*. (International Conference on Learning Representations, 2015).
42. Zeiler, M. D. Adadelta: an adaptive learning rate method. Preprint at https://doi.org/10.48550/arXiv.1212.5701 (2012).

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62573345 & 62273273).

Author contributions

Y. L.: Data curation, Methodology, Software, Validation, Writing—original draft. E. Z.: Supervision, Project administration, Methodology, Writing—review & editing. G.L.: Data curation. J. D.: Data curation.

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to Erhu Zhang.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025