Unfettered Ink: Restoring Legibility and Stylistic Consistency in Immersive Air Handwriting

A. APPENDIX

A.1. Handwriting Dataset

CASIA-OLHWDB CASIA-OLHWDB (1.0-1.2)[1] is an online handwritten Chinese character database collected by the Institute of Automation, Chinese Academy of Sciences between 2007 and 2010. The database contains handwritten single-character text samples from 1,020 writers. During the data collection process, an Anoto digital pen and specially designed dot matrix paper were used. Writers wrote on the dot matrix paper, and the Anoto pen recorded the trajectory of the pen tip in real time, including the start, end, and trajectory coordinate information of the strokes.

IAHCT-UCAS2018[2] is the first in-air handwritten Chinese text dataset, which contains 15,671 textlines of 260,224 characters. The whole dataset is split into the training set and test set, where the data of 277 writers (11,807 lines with 196,129 characters of 3564 classes) is used for training and the remaining 92 writers (3864 lines with 64,095 characters of 2397 classes) for testing.

In our approach, the quality of character reconstruction relies on the degree of matching between the target character and the style reference character. Several studies have released datasets for in-air handwriting fonts [2], but these datasets are primarily used for in-air handwriting character recognition tasks.

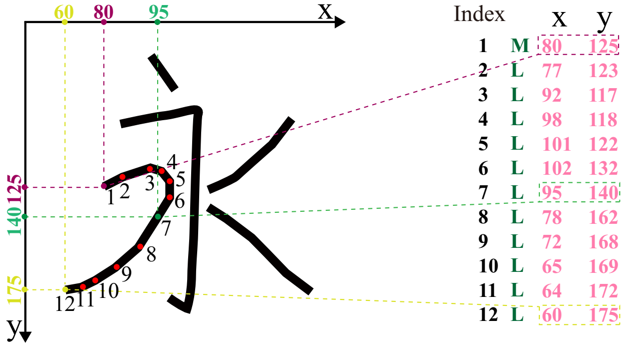
Our proposed method falls under supervised learning; however, other datasets do not provide effective target characters. Therefore, we select CASIA-OLHWDB (1.0–1.2) [1] as both the style reference and the target character dataset. In addition, we employ IAHCT-UCAS2018 [2] as the dataset of in-air handwritten characters serving as model inputs. The model disentangles ”content” and ”style” in the feature space, thereby enabling cross-style handwritten character optimization while preserving the semantic integrity of the characters.

A.2. Data Structure

In order to extract the details of the writing process and maintain the editability of the handwritten characters, we treat the user's handwritten characters as vector images composed of a series of straight lines, and represent these characters using SVG vector drawing parameters. As shown in Figure 1(a) and (b), specifically, the command M (MoveTo) indicates that the user stops writing and moves to a new position in the air, while the command L (LineTo) indicates the movement path when the user writes in the air. Each handwriting trajectory is composed of L drawing parameters V, where each drawing parameter $v_i = (h_i, p_i)$ includes the drawing command type $h_i \in \mathrm{M}, \mathrm{L}$ and the corresponding drawing coordinate $p_i = (x, y)$.

A.3. Handwriting Resampling

In order to map the handwritten trajectory from the time domain to the frequency domain, we perform linear interpolation on the trajectory data to ensure that handwritten trajectories of varying complexity have the same number of sample points. Statistical analysis shows that 256 sample points are sufficient to represent all characters in the dataset.



Index		x	y
1	M	80	125
2	L	77	123
3	L	92	117
4	L	98	118
5	L	101	122
6	L	102	132
7	L	95	140
8	L	78	162
9	L	72	168
10	L	65	169
11	L	64	172
12	L	60	175

(a) Handwritten Character

(b) Drawing Parameters

Fig. 1. (a) Handwritten character. (b) SVG representation of a handwritten character.

As shown in algorithm 1, the goal of this algorithm is to resample a series of SVG drawing parameters into a specified number of line segments, thereby achieving a uniform distribution of the path while preserving its original shape. First, the input parameters $V = v_1, v_2, \ldots, v_N$ are segmented into several sub-paths according to the M commands, forming the set S; next, the total number of original line segments is calculated based on the number of line segments in each sub-path, and the target number of line segments $256 - |S|$ is proportionally allocated to each sub-path; subsequently, for each sub-path, a new coordinate sequence is uniformly sampled along the original path using cumulative distance and linear interpolation methods, thereby obtaining the interpolated sub-path; finally, these sub-paths are converted back into SVG commands.

Additionally, we eliminate the discrete representation introduced by the drawing commands. We insert a $(0, 0)$ coordinate before each drawing command M as a marker for stroke segmentation and remove the drawing command M, as shown in Figure 2.

A.4. Isometric Embedding and De-Embedding

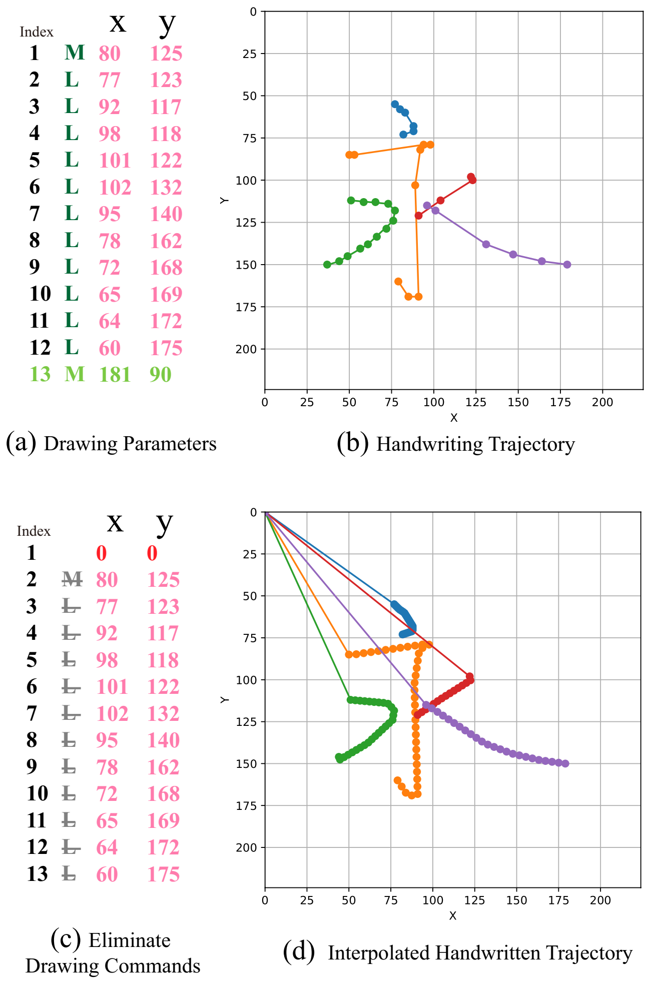
To enhance the network's ability to express frequency domain features while strictly preserving the energy conservation property of the Fourier transform, we have constructed a unitary mapping on the Stiefel manifold. Let the original two-dimensional complex frequency spectrum vector be:

$$\mathcal{Z}[k] = \begin{bmatrix} Z_x[k] \\ Z_y[k] \end{bmatrix} \in \mathbb{C}^2, k = 0, \ldots, L-1. \tag{1}$$

Select a column-orthogonal matrix $A \in \mathbb{C}^{D \times 2}$ that satisfies $A^H A = I_2$:

Algorithm 1: Minimal Interpolation Algorithm

```
Input: $V = \{v_1, \ldots, v_N\}$: A sequence of SVG
       parameters, each $v_i = (h_i, p_i)$, $h_i \in \{$"M", "L"$\}$,
       $p_i = (x_i, y_i)$
Output: $\hat{V}$: Interpolated SVG parameters
Function Main():
    $S \leftarrow \emptyset$
    $P \leftarrow \emptyset$
    for $i \leftarrow 1$ to $N$ do
        if $h_i =$ "M" $\wedge$ $|P| > 0$ then
            $S \leftarrow S \cup \{P\}$;   $P \leftarrow \emptyset$
        end
        $P \leftarrow P \cup \{p_i\}$
    end
    if $|P| > 0$ then
        $S \leftarrow S \cup \{P\}$
    end
    $K \leftarrow 256 - |S|$
    foreach Sub-path $P$ in $S$ do
        $t \leftarrow \mathrm{round}\left(K \cdot \frac{|P|-1}{\sum(|P_j|-1)}\right)$
    end
    $\hat{S} \leftarrow \emptyset$
    foreach Sub-path $P$ in $S$ do
        $\hat{P} \leftarrow$ Interpolate $P$ to $t+1$ points
        $\hat{S} \leftarrow \hat{S} \cup \{\hat{P}\}$
    end
    $\hat{V} \leftarrow \emptyset$
    foreach $\hat{P}$ in $\hat{S}$ do
        $\hat{V} \leftarrow \hat{V} \cup \{($"M", $\hat{P}[0])\}$
        for $j \leftarrow 1$ to $|\hat{P}| - 1$ do
            $\hat{V} \leftarrow \hat{V} \cup \{($"L", $\hat{P}[j])\}$
        end
    end
return $\hat{V}$
```



Index		x	y
1	M	80	125
2	L	77	123
3	L	92	117
4	L	98	118
5	L	101	122
6	L	102	132
7	L	95	140
8	L	78	162
9	L	72	168
10	L	65	169
11	L	64	172
12	L	60	175
13	M	181	90

(a) Drawing Parameters

(b) Handwriting Trajectory

Index		x	y
1		0	0
2	~~M~~	80	125
3	~~L~~	77	123
4	~~L~~	92	117
5	~~L~~	98	118
6	~~L~~	101	122
7	~~L~~	102	132
8	~~L~~	95	140
9	~~L~~	78	162
10	~~L~~	72	168
11	~~L~~	65	169
12	~~L~~	64	172
13	~~L~~	60	175

(c) Eliminate Drawing Commands

(d) Interpolated Handwritten Trajectory

Fig. 2. (a) Original handwritten trajectory drawing commands. (b) Original handwritten trajectory. (c) Eliminate drawing commands. (d) Interpolated handwritten trajectory.

$$A = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ \vdots & \vdots \\ a_{D1} & a_{D2} \end{pmatrix} \in \mathbb{C}^{D\times 2}, \qquad A^H A = I_2, \tag{2}$$

its two column vectors lie on the Stiefel manifold $\mathrm{St}(2, D)$, forming an orthogonal unitary basis in $\mathbb{C}^D$. For each frequency index k, through:

$$\hat{\mathcal{Z}}[k] = A\,\mathcal{Z}[k] \in \mathbb{C}^D. \tag{3}$$

This allows the original spectral matrix to be mapped from $\mathbb{C}^{L\times 2}$ to $\mathbb{C}^{L\times D}$. In the time domain, this mapping corresponds to:

$$\widetilde{x}_d[t] = \sum_{c=1}^{2} h_{dc}\, x_c[t], \quad h_{dc} = \frac{1}{L}\sum_{k=0}^{L-1} A_{dc}\, e^{i2\pi kt/L}, \tag{4}$$

where filter kernel h_{dc} is independent of time t, which is equivalent to performing D parallel distortion-free filtering channels. This not only expands the feature channels available to the network but also ensures full compatibility with traditional frequency domain operations such as IDFT and bandpass filtering. Since $A^H A = I_2$, this mapping is not only linearly invertible but also strictly preserves the Parseval energy of the original frequency spectrum. The original signal can be accurately recovered through $\mathcal{Z}[k] = A^H\,\hat{\mathcal{Z}}[k] \in \mathbb{C}^2$.

A.5. Visualization of Isometric Embedding

As shown in Figure 3, we construct a two-channel composite signal of length $L = 128$ and plot their time-domain waveforms.

$$x_1[t] = \sin\bigl(2\pi \cdot 5\,t/L\bigr), \tag{5}$$

$$x_2[t] = \cos\bigl(2\pi \cdot 10\,t/L\bigr). \tag{6}$$

As shown in Figure 4, we perform FFT on the above signals, obtaining significant peaks at $k = 5$ for $Z_1[k]$ and at $k = 10$ for $Z_2[k]$, which visually demonstrates the frequency components of the signals.

We use a fixed unitary mapping to elevate the frequency domain features to $D = 4$:

$$A_{\text{fixed}} = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{pmatrix}, \quad A_{\text{fixed}}^H A_{\text{fixed}} = I_2, \tag{7}$$

We compute $\hat{Z}[k] = A_{\text{fixed}} Z[k]$ and display the magnitude spectrum of each channel using four curves $(d = 1, \ldots, 4)$ in a 3D plot. As shown in Figure 5, the original frequency spectrum peaks are fully retained in each mapped dimension, confirming the energy conservation of the mapping.

Additionally, we perform a QR decomposition on a randomly generated 4×2 matrix and take its column orthogonal basis A_{rand}. We then compute $\hat{Z}_{\text{rand}}[k] = A_{\text{rand}} Z[k]$. As shown in Figure 6, all dimensions still preserve the original two-channel peaks.

We use the inverse transformation on the randomly generated A_{rand} to obtain the time-domain image, as shown in Figure 7. Through the unitary mapping on the Stiefel manifold, we can extend the two-dimensional frequency spectrum to D-dimensional space without redundancy, while strictly satisfying Parseval's energy conservation and maintaining linear invertibility. Whether using a fixed basis or a random basis, the mapped multi-channel frequency spectrum can fully reproduce the original frequency components, providing theoretical and visual support for the frequency domain feature expansion in models like neural networks.

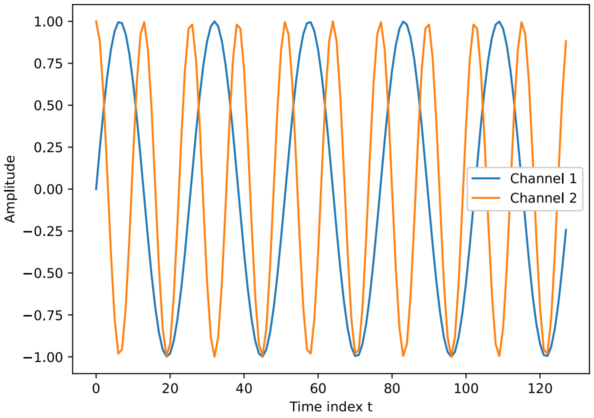


Fig. 3. Two-channel time-domain signals.

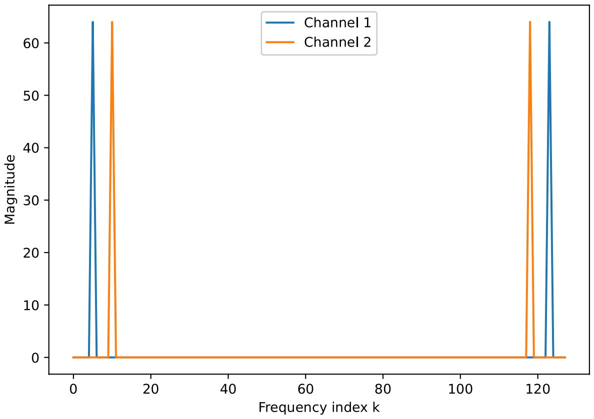


Fig. 4. Original frequency-domain magnitude.

A.6. Ablation Study on Isometric Embedding

To evaluate the effectiveness of isometric embedding, we compared three schemes: ordinary linear dimensionality expansion (without orthogonal constraints), fixed orthogonal unitary basis A , and learnable orthogonal unitary basis A.

As shown in Table 1, replacing ordinary linear dimensionality expansion (Linear) with a fixed orthogonal unitary mapping (Fixed-A) resulted in an increase of approximately 3.5 points for Style Score and 1.9 points for Content Score, while the DTW error decreased by 12.4%. This demonstrates that energy conservation and invertible mapping significantly improved generation quality and temporal consistency. By maintaining the $A^H A = I_2$ constraint, further making A learnable (Learnable-A) provided an additional 2.5 points (Style) and 2.3 points (Content) improvement. The model can automatically adjust the two basis vectors to better align with the dominant directions in the data distribution, thus achieving stronger generation capability while preserving energy conservation and invertibility.

As shown in Table 2, orthogonal unitary bases not only eliminate

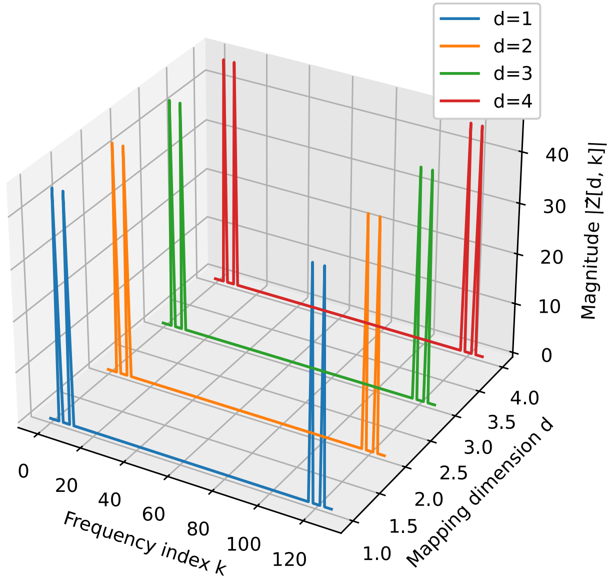


Fig. 5. Mapping of the frequency domain magnitude spectrum using orthogonal unitary bases.

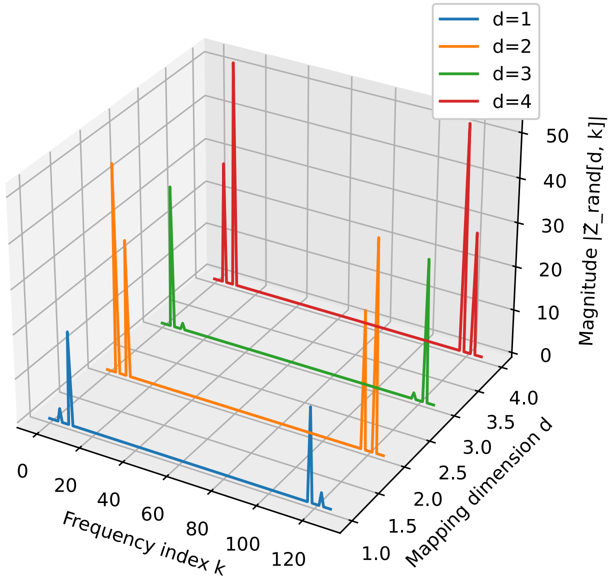


Fig. 6. Mapping of the original frequency domain to a higher dimension using the column orthogonal matrix obtained from the QR decomposition of a random matrix, and plotting the resulting magnitude spectrum.

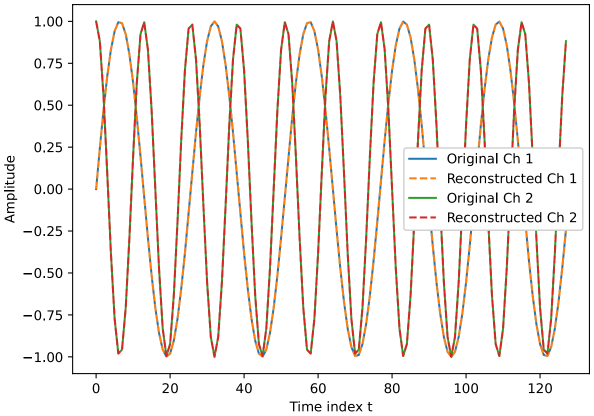


Fig. 7. Reconstructed time-domain signals.

Method	Style ↑	Content ↑	DTW ↓
Linear (non-orthogonal)	82.15	88.41	1.0627
Fixed-A (orthogonal)	85.92	90.27	0.9343
Learnable-A (orthogonal)	88.12	92.23	0.8632

Table 1. Ablation study on the Stiefel-manifold isometric embedding. ”Linear” refers to linear dimensionality expansion; ”Fixed-A” uses a predefined column orthogonal matrix A; ”Learnable-A” imposes a unitary constraint on A and learns it end-to-end.

the energy bias introduced by linear dimensionality expansion but also provide invertible, interpretable, and easily integrable channel transformations with traditional frequency domain operations. On top of this, making the basis vectors learnable further unleashes the model’s capacity and adaptability, leading to the optimal generation performance.

A.7. Low-pass Filtering Theory

We assume that in the frequency-domain representation of the handwritten trajectory, the shape of the trajectory is determined by the low-frequency components, while the writing jitter manifests as high-frequency noise. This jitter can be suppressed through a filter without damaging the character shape information. We first prove from a theoretical perspective that the writing jitter manifests as high-frequency noise, which can be suppressed by the filter.

Let the ideal handwritten trajectory be a two-dimensional smooth curve $s(t) = \big(s_x(t), s_y(t)\big), t \in [0, T]$, and let the jitter caused by writing be $n(t)$. After discretely sampling with a period of Δ, , we obtain a sequence of length L given by $X[\ell] = s(\ell\Delta) + n(\ell\Delta)$. The Discrete Fourier Transform (DFT) of the sequence is defined as:

$$\mathcal{Z}[k] = \sum_{\ell=0}^{L-1} X[\ell] \cdot e^{-i2\pi k\ell/L}, k = 0, 1, \ldots, L-1, \qquad (8)$$

where $\mathcal{Z}[k] \in \mathbb{C}^2$ is the frequency-domain component represented in the complex domain, and k is the frequency index. Since the DFT is a linear operator, we have $\mathcal{Z}[k] = \mathcal{S}[k] + \mathcal{N}[k]$. Let there be a

Aspect	**Linear Mapping**	**Orthogonal Unitary Mapping (Ours)**
Energy preservation	Amplification / attenuation	Parseval equality strictly holds
Invertibility	Rank-dependent	A^{H} is exact inverse
Parameter count	Low	Comparable, with unitary constraint
Training stability	Possible gradient explosion / vanishing	Spectral norm constant, smoother gradients
Channel interpretability	Hard to interpret	Basis vectors orthogonal, non-interfering
Compatibility with frequency-domain ops	Requires re-calibration	Fully compatible with IDFT / band-pass filters

Table 2. Advantages over plain linear lifting.

cutoff index $k_c \ll L$ such that:

$$\sum_{|k|\le k_c}|\mathcal{S}[k]|^2 \gg \sum_{|k|\le k_c}|\mathcal{N}[k]|^2,\ \sum_{|k|>k_c}|\mathcal{N}[k]|^2 \gg \sum_{|k|>k_c}|\mathcal{S}[k]|^2. \qquad (9)$$

According to the spectrum decay lemma for smooth functions (A.9), when $|k| > k_c$, we have $|\mathcal{S}[k]|^2 \le C|k|^{-2p}$, therefore:

$$\sum_{|k|>k_c}|\mathcal{S}[k]|^2 \le \tfrac{C}{2p-1}\,k_c^{-(2p-1)} \to 0. \qquad (10)$$

This indicates that only a limited number of low-frequency components are needed to accurately reconstruct the original signal, while the high-frequency components can be regarded as noise disturbances and filtered out. If k_c is known accurately, the simplest ideal low-pass window function is given by:

$$W^\star[k] = \begin{cases} 1, & |k| \le k_c, \\ 0, & |k| > k_c. \end{cases} \qquad (11)$$

After filtering, the reconstruction of the signal is given by $\widehat{\mathcal{S}}[k] = W^\star[k]\mathcal{Z}[k]$. The mean squared reconstruction error can be decomposed into the loss of the high-frequency components of the signal and the residual low-frequency noise as follows:

$$\mathbb{E}[\|\widehat{S} - S\|_2^2] = \sum_{|k|>k_c}|\mathcal{S}[k]|^2 + \sum_{|k|\le k_c}\mathbb{E}\big[|\mathcal{N}[k]|^2\big], \qquad (12)$$

where the first term decays as a power law with increasing k_c, while the second term represents the residual noise in the low-frequency region. If any components with $|k| > k_c$ are included within the passband, high-frequency noise will be introduced, increasing the error. Therefore, the ideal low-pass filter $W^\star$ is the solution that minimizes the mean squared error between the reconstructed signal $\widehat{S}$ and the original signal S.

In the frequency-domain data of handwritten trajectories, each writer's writing trajectory has its own variations, making it difficult to explicitly define k_c. Additionally, some noise still remains in the low-frequency region, which is difficult to filter out using a low-pass window function. To balance the generality and individuality of handwritten trajectories, we propose **Static Filters** and **Dynamic Filters**, which adaptively learn a set of complex gating coefficients to approximate an ideal window specific to the sample.

A.8. Frequency Domain Style Encoder

Handwritten trajectories are periodic and regular spatial motion signals, and their frequency domain features can reflect changes in writing speed and style, as well as reveal details that are difficult to perceive in the time domain. Inspired by the properties of Fourier amplitude and phase, we propose a Fourier-domain style fusion strategy that maintains the phase component unchanged while fine-tuning the amplitude component to adjust style characteristics, ensuring that the structure and semantics of the trajectory remain unaffected.

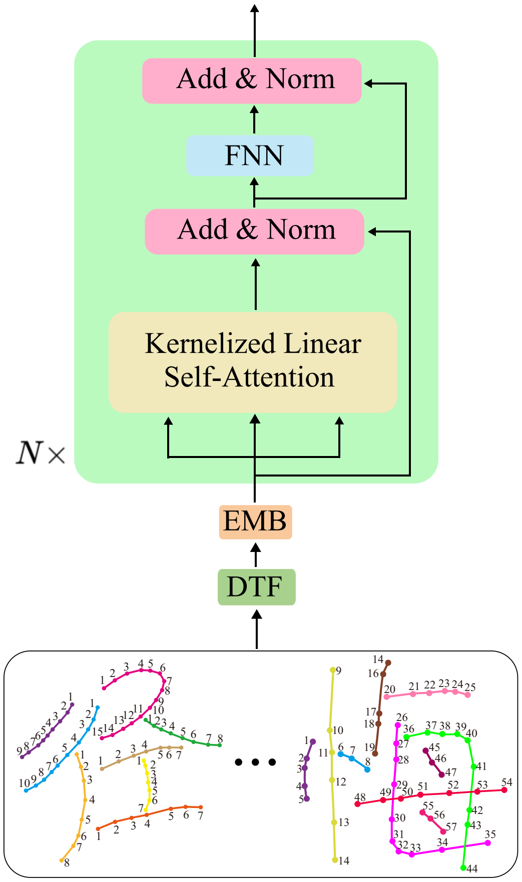


Fig. 8. Frequency domain style encoder architecture.

To inject style features from the sequence modality into the stylized reconstruction process, we finely perturb the amplitude component to modulate the style. As shown in Figure 8, we introduce a Kernelized Linear Self-Attention-based complex frequency domain style encoder.

Specifically, in each training iteration, we first randomly select K coordinate sequences of handwritten characters as style references. Then, each sequence is uniformly resampled to a fixed number of L points. Next, the resampled sequences undergo a discrete Fourier transform, mapping them to the frequency domain. To enhance the representation of frequency spectrum features, we apply Isometric Embedding to project each frequency domain representation into a higher-dimensional space. Finally, the style encoder consists of three stacked complex frequency domain self-attention modules that process these high-dimensional embeddings to extract the final style feature vector$\mathcal{F}_s = \mathcal{A}\left(F_s\right) \cdot e^{j\mathcal{P}(F_s)}$.

A.9. Spectral Decay Lemma for Smooth Functions

Let the continuous function $s(t)$ be at least p -times continuously differentiable on the interval $[0, T]$ i.e., $s \in C^p$. Then its Fourier coefficients are given by:

$$s(k) = \frac{1}{T} \int_0^T s(t)\, e^{-j2\pi kt/T}\, dt, \qquad k \in \mathbb{Z}. \tag{13}$$

By performing p successive integrations by parts on the above expression and using the periodic boundary terms to cancel each other out, we obtain:

$$\hat{s}(k) = \frac{1}{(j2\pi k/T)^p} \cdot \frac{1}{T} \int_0^T s^{(p)}(t)\, e^{-j2\pi kt/T}\, dt. \tag{14}$$

From $\|s^{(p)}\|_\infty \;\le\; M$, it follows that there exists a constant$C_0 := \frac{M\,T^p}{(2\pi)^p}$ such that:

$$|\hat{s}(k)| \;\le\; \frac{M\,T^p}{(2\pi)^p}\, |k|^{-p}, \qquad |k| \to \infty. \tag{15}$$

Taking the square of equation 15, we obtain:

$$|\hat{s}(k)|^2 \;\le\; C\, |k|^{-2p}, \quad C := C_0^2. \tag{16}$$

Let the cutoff frequency be $k_c \in \mathbb{N}$. The tail energy of the frequency spectrum is given by:

$$\sum_{|k|>k_c} |\hat{s}(k)|^2 \;\le\; 2C \sum_{k=k_c+1}^{\infty} k^{-2p}. \tag{17}$$

Since $f(x) = x^{-2p}$ is monotonically decreasing for $x \ge 1$ and $2p > 1$, we can estimate using the integral comparison test:

$$\sum_{k=k_c+1}^{\infty} k^{-2p} \;\le\; \int_{k_c}^{\infty} x^{-2p}\, dx \;=\; \frac{k_c^{-(2p-1)}}{2p-1}. \tag{18}$$

Substituting this into the previous expression and combining constants, with $C' = \dfrac{2C}{2p-1}$, we obtain:

$$\sum_{|k|>k_c} |\hat{s}(k)|^2 \le \frac{C}{2p-1} k_c^{-(2p-1)} \qquad (k_c \to \infty). \tag{19}$$

If s is at least p-times smooth, its Fourier amplitude decays according to the $|k|^{-p}$ rate.

From the above proof, it can be concluded that the smoother the curve, the slower its local variations. In the frequency domain, such slow variations can be effectively approximated using a small number of low-frequency sine waves, while the energy of high-frequency components can be almost neglected.

In handwritten trajectories, the motion involved in human writing typically exhibits coherent and smooth characteristics, with the trajectory function often being continuous across several derivatives. Therefore, most of the energy is concentrated in the low-frequency region, and only a small number of low-frequency Fourier coefficients are needed to adequately reconstruct the original trajectory. In contrast, the jitter noise generated during writing usually manifests as high-frequency, small-amplitude random disturbances, which have poor smoothness and their energy is primarily distributed in the high-frequency part.

Thus, in frequency domain processing, we only need to identify an appropriate decay threshold k_c, retain the low-frequency coefficients satisfying $|k| \le k_c$, and attenuate or directly discard the high-frequency components. This allows us to effectively smooth the trajectory without compromising its overall shape characteristics. This principle not only forms the theoretical foundation for frequency domain filtering methods but also serves as the core basis for suppressing jitter through filters.

A.10. Frequency-Domain Filtering Visualization

We hypothesize that in the frequency domain representation of handwritten trajectories, the shape of the trajectory is determined by low-frequency components, while the jitter of the writing is represented as high-frequency noise. This writing jitter can be suppressed by a filter without destroying the shape information of the characters. In the previous section, we validated this hypothesis theoretically. To further verify it, we add random jitter to the original handwritten trajectory (Figure 9(a)) and normalize the trajectory coordinates to the range $[-1, 1]$ (Figure 9(b)). We then perform a discrete Fourier transform (DFT) on the jittered trajectory, separately on the $s_x[n]$ and $s_y[n]$ sequences, obtaining the complex spectra $\{\mathcal{Z}_x[k], \mathcal{Z}_y[k]\}$.

We choose two different cutoff frequencies, $k_c = 5$ and $k_c = 10$, and set the high-frequency components from $\{k_{c+1}, \ldots, N - k_{c-1}\}$ to zero, while retaining the frequencies $\{0, \ldots, k_c\}$ and the symmetric negative frequencies. After filtering the spectra $\{\mathcal{Z}_x[k], \mathcal{Z}_y[k]\}$, we perform an inverse discrete Fourier transform (IDFT), and map the result back to the $[-1, 1]$ interval, obtaining Figures 9(c) and (d). When $k_c = 5$ (Figure 9(c)), only the lowest 6 positive and negative frequency components are retained. The trajectory is relatively smooth, and most of the jitter has been removed. When $k_c = 10$ (Figure 9(d)), more middle and low-frequency components are retained, leading to the "reintroduction" of more jitter. Additionally, at $k_c = 5$, some noise still remains in the low-frequency region, which is difficult to filter out using a low-pass window function.

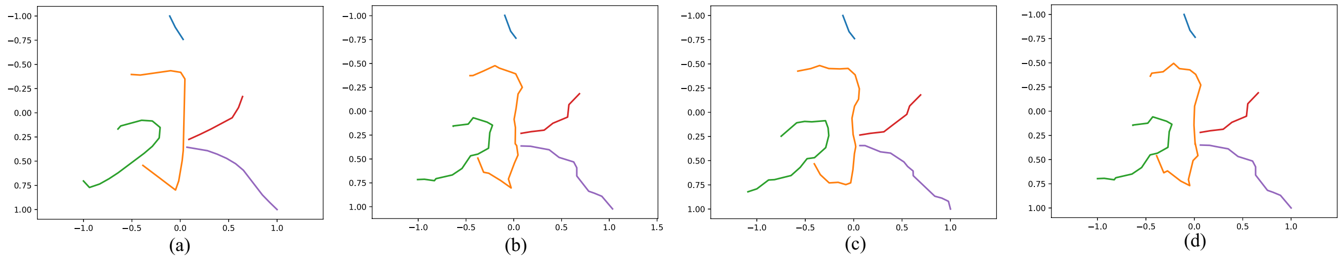
As shown in Fig. 10, for each frequency index k, we reconstruct a sine wave in the x-direction and y-direction based on its magnitude and phase. This reconstructed sine wave represents the contribution of the k-th DFT basis function to the handwriting trajectory in the time domain. When $k = 1$ and $k = 2$, the weighted sine waves have the largest amplitudes, indicating that most of the shape information of the handwriting trajectory is concentrated in the low-frequency components. When $k > 3$, the amplitudes of the components decrease sharply, showing that their contributions to the overall stroke shape are negligible. Although the higher-order components are small, they are not exactly zero and their waveforms appear irregular. This behavior reflects spectral leakage or the superposition of random jitter noise.

A.11. Kernelized Linear Self-Attention

Given $Q, K, V \in \mathbb{R}^{L \times d}$, the conventional self-attention is represented as:

$$\mathrm{Att}(Q, K, V) = \mathrm{softmax}\left(\frac{1}{\sqrt{d}} Q K^\top\right) V \tag{20}$$

If we use a general similarity function sim, the above expression can be expanded in a per-position form as:



(a) (b) (c) (d)

Fig. 9. (a) Original handwritten trajectory. (b) Handwritten trajectory with added noise. (c) Reconstructed handwritten trajectory using the first $k = 5$ frequency components. (d) Reconstructed handwritten trajectory using the first $k = 10$frequency components.

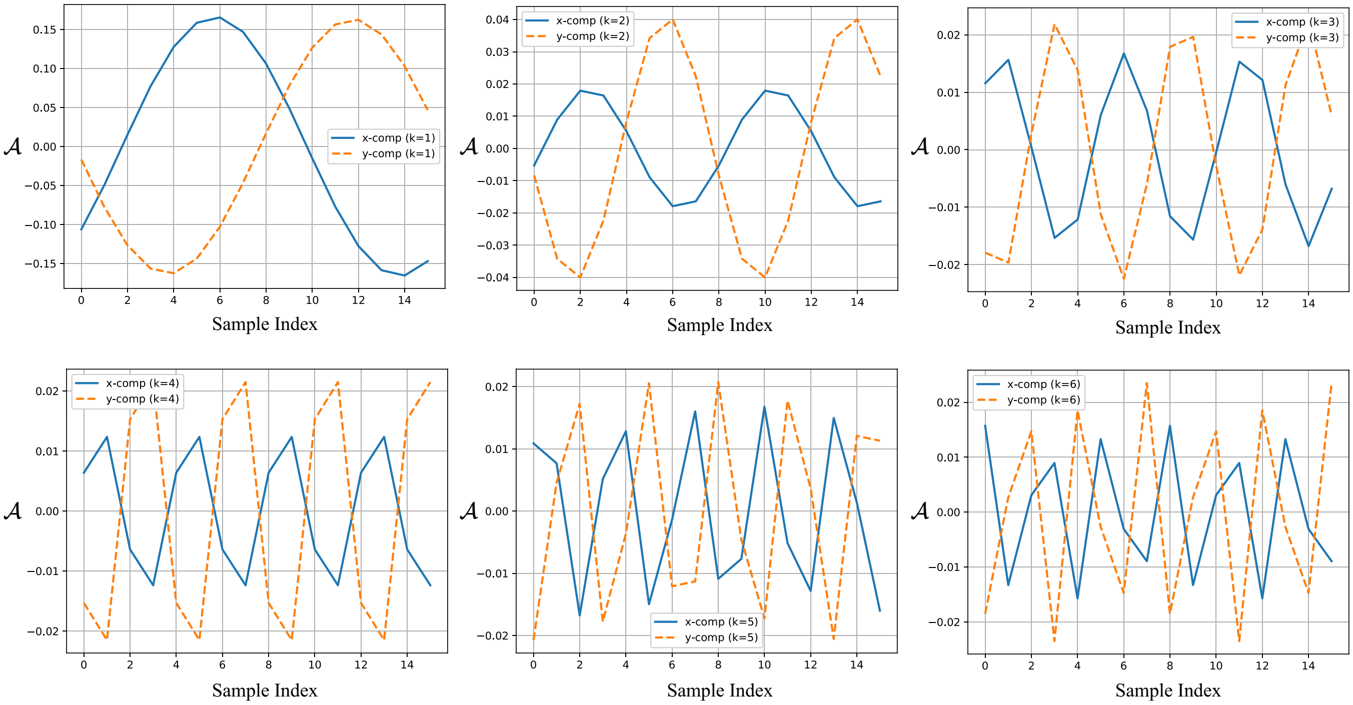


Fig. 10. Weighted sine waves of each frequency component of the handwritten trajectory in the time domain.

$$V_i' = \frac{\sum_{j=1}^{L} \mathrm{sim}(Q_i, K_j) V_j}{\sum_{j=1}^{L} \mathrm{sim}(Q_i, K_j)}. \quad (21)$$

When $\mathrm{sim}(q,k) = \exp(qk^\top/\sqrt{d})$), it is consistent with the softmax attention in self-attention. If the kernel function satisfies $k(x,y) = \phi(x)^\top \phi(y)$, then we have:

$$V_i' = \frac{\sum_{j=1}^{L} \phi(Q_i)^\top \phi(K_j) V_j}{\sum_{j=1}^{L} \phi(Q_i)^\top \phi(K_j)}. \quad (22)$$

Using the associativity of the inner product, we first aggregate over all keys as follows:

$$Z = \sum_{j=1}^{L} \phi(K_j) \in \mathbb{R}^m, \quad S = \sum_{j=1}^{L} \phi(K_j)\, V_j^\top \in \mathbb{R}^{m\times d}, \quad (23)$$

Then, for each query Q_i, only two m-dimensional inner products are needed:

$$V_i' = \frac{\phi(Q_i)^\top\, S}{\phi(Q_i)^\top\, Z}. \quad (24)$$

This reduces the time complexity from $\mathcal{O}(N^2)$ to $\mathcal{O}(Nm)$.

For the translation-invariant kernel $k(x,y) = \exp(-\|x-y\|^2/(2\sigma^2))$, the Bochner theorem guarantees:

$$k(x,y) = E_{\omega\sim\mathcal{N}(0,\sigma^{-2}I)}\big[e^{i\,\omega^\top(x-y)}\big]. \quad (25)$$

By using Monte Carlo sampling for $\{\omega_i\}_{i=1}^m$ and $\{b_i\}_{i=1}^m$, we construct a real-valued mapping:

$$\phi(x) = \sqrt{\frac{2}{m}}\,\Big[\cos\big(\omega_1^\top x + b_1\big),\ \ldots,\ \cos\big(\omega_m^\top x + b_m\big)\Big]^\top, \quad (26)$$

which satisfies:

$$\begin{aligned}\phi(x)^\top\phi(y) &= \frac{2}{m}\sum_{i=1}^{m}\cos\big(\omega_i^\top x + b_i\big)\ \cos\big(\omega_i^\top y + b_i\big) \\ &\approx k(x,y).\end{aligned} \quad (27)$$

In this way, we retain the similarity measure of the kernel function while linearizing the computational cost.

A.12. Evaluation Metrics

DTW: Dynamic Time Warping (DTW) [3] is used to compute the distance between two sequences of different lengths. Therefore, we use DTW to evaluate the similarity between real and generated handwriting, where a lower DTW value indicates higher similarity.

In addition, to quantitatively evaluate our method from both content and style perspectives, we adopt the evaluation approach proposed in methods such as WriteLikeYou [4], SDT [5], and ElegantlyWritten [**?**], which score the generated handwriting using content and style classifiers.

Content Score: In this study, we use the CASIA-OLHWDB (1.0–1.2) [1] online handwritten Chinese-character database, which was provided by 420 writers and covers 3,755 common character classes, with a total of over one million samples. First, each stroke trajectory in the CASIA-OLHWDB1.1 training set is rendered as a 256×256-pixel grayscale image, and then the data are randomly split into training and test sets at a 90 % : 10 % ratio. Based on a ResNet-50 network pretrained on ImageNet, we fine-tune the model using the Adam optimizer ($\beta_1 = 0.9$, $\beta_1 = 0.999$) with an initial learning rate of 0.001, a weight-decay of 1×10^{-4}, and a batch size of 256. Training is set to run for a maximum of 500 epochs, but an Early Stopping strategy is enabled: we monitor the validation-set loss in real time and automatically terminate training (restoring the model weights from the epoch with the lowest validation loss) if the loss does not decrease for 20 consecutive epochs. Experimental results show that the fine-tuned model achieves a character-recognition accuracy of 99.0 % on the test set.

To more precisely evaluate the content consistency between generated characters and real ones, we take the model's predicted probability for the true character class as the content-consistency score. From all character classes, we randomly sample 200 classes for evaluation, average their individual scores, and thereby obtain the final mean content-consistency score.

Style Score: Similar to the Content Score evaluation, we trained a writer-recognition classifier on the test split of the CASIA-OLHWDB (1.0–1.2) dataset [1] to define the Style Score. This classifier achieves a recognition accuracy of 96%. We take the classifier's predicted probability for the true writer as the style-consistency score, then average those scores over 200 characters to obtain the final mean Style Score.

To further validate the effectiveness of the Style Score metric, we conducted a writer-identification experiment on the CASIA-OLHWDB (1.0–1.2) dataset with 40 volunteers. Each volunteer was asked to identify the true author of 10 handwritten characters from each of 100 different writers. The volunteers achieved an accuracy of 86%, significantly lower than the 96% accuracy of our ResNet-50 model. This result further confirms the reliability of the Style Score in capturing handwriting style consistency.

User Prefer.: We invited 40 volunteers to subjectively evaluate the 200 characters generated by each method. During the evaluation, volunteers were shown a reconstructed character alongside its original target; if the reconstruction exhibited no significant difference from the target font, it was deemed "correct." Volunteers can zoom in on the reconstruction results to view the details. Finally, we used the mean number of "correct" characters per method as our quantitative evaluation metric. When the reconstructed output exactly matches the target, the User Prefer. metric reaches 100%. This metric is computed independently for each method, in line with the approach used in Elegantly Written[**?**]. Independent computation simplifies the formulation of statistical hypotheses and the construction of confidence intervals; by contrast, within-group comparisons introduce additional preference-contrast bias due to competition among samples.

A.13. Effectiveness of Kernelized Linear Self-Attention

To verify the ability of Kernelized Linear Self-Attention (KLSA) to reduce computational overhead while maintaining modeling performance, we conducted a comparative experiment between KLSA and Complex Frequency Domain Multi-Head Self-Attention (CFD-MHSA Appendix A.13.1) on the same dataset. Table 3 lists the performance of both methods in terms of Style Score, Content Score, DTW similarity, and per-inference forward pass time (Inference Time).

From Table 3, we observe that KLSA's style and content quality are almost identical to those of CFD-MHSA, with a slight advantage in the DTW metric. Meanwhile, KLSA's per-inference forward pass time is approximately 95 ms, about 37 % lower than CFD-MHSA's 150 ms. Theoretically, CFD-MHSA relies on frequency-domain convolution with $\mathcal{O}(L\log L)$ complexity and a high constant factor, whereas KLSA uses kernel mapping to reduce attention com-

Evaluation Metric	CFD-MHSA	KLSA (ours)
Style Score ↑	88.95	88.35
Content Score ↑	92.45	92.64
DTW ↓	0.8775	0.8745
Inference Time (ms) ↓	150	95

Table 3. Performance and Speed Comparison between KLSA and CFD-MHSA.

putation to two m-dimensional inner products, achieving $\mathcal{O}(Lm)$ complexity. In summary, our method delivers comparable modeling performance to CFD-MHSA while offering lower time complexity and faster inference speed.

A.13.1. Complex Frequency Domain Multi-Head Self-Attention.

For the input complex feature $\mathcal{X} = [x_1, x_2, \ldots, x_L] \in \mathbb{C}^{L\times D}$, each element is a complex number, and each component can be represented as $x_d = a + i, b \in \mathbb{C}^{1\times D}$. Through a complex linear mapping, we obtain the query $\mathcal{Q}$, key $\mathcal{K}$, and value $\mathcal{V}$:

$$\mathcal{Q} = \mathcal{W}_q \cdot \mathcal{X}, \quad \mathcal{K} = \mathcal{W}_k \cdot \mathcal{X}, \quad \mathcal{V} = \mathcal{W}_v \cdot \mathcal{X}, \tag{28}$$

where $\mathcal{W}_q, \mathcal{W}_k, \mathcal{W}_v \in \mathbb{C}^{D\times M}$ are complex weight matrices. Specifically, for $x_d = a_d + i\,b_d$ and $w_d = p_d + i\,q_d$, the corresponding product is computed by combining the real and imaginary parts as follows:

$$(p_d + i\,q_d)(a_d + i\,b_d) = (p_d\,a_d - q_d\,b_d) + i\,(p_d\,b_d + q_d\,a_d). \tag{29}$$

By mapping the D-dimensional features to M dimensions in the manner described above, both the real and imaginary parts of the information are retained. After obtaining the complex query $\mathcal{Q}$ and complex key $\mathcal{K}$, we need to compute attention in the complex domain to generate the appropriate weighted sum for each query. Since directly computing the dot product of complex vectors results in complex values, which do not satisfy the requirement that inputs for the softmax normalization be real numbers, we adopt the following method. First, we compute the inner product between each query vector $q_{i,m} = [q_{i,1}, q_{i,2}, \ldots, q_{i,M}]$ and each key vector $k_{k,m} = [k_{j,1}, k_{j,2}, \ldots, k_{j,M}]$. When calculating the inner product, we introduce the conjugate operation:

$$\langle q_i, k_j \rangle = \sum_{m=1}^{M} q_{i,m}\,\overline{k}_{j,m}. \tag{30}$$

In the complex domain, to ensure that the resulting attention scores are real numbers, we use the magnitude of the inner product between complex vectors as the measure of attention similarity:

$$\alpha_{ij} = \sum_{m=1}^{M} (a_{i,m}c_{j,m} + b_{i,m}d_{j,m}), \tag{31}$$

$$\beta_{ij} = \sum_{m=1}^{M} (b_{i,m}c_{j,m} - a_{i,m}d_{j,m}), \tag{32}$$

$$s_{ij} = \sqrt{\alpha_{ij}^2 + \beta_{ij}^2}. \tag{33}$$

To convert these scores into a probability distribution, we compute the softmax normalization for each query q_i to obtain the attention weights:

$$\alpha_{ij} = \frac{\exp(\frac{s_{ij}}{\sqrt{M}})}{\sum_{k=1}^{N} \exp(\frac{s_{ik}}{\sqrt{M}})}. \tag{34}$$

Finally, these attention weights are applied to the complex values $\mathcal{V}$ to obtain the weighted sum representation for each query:

$$z_i = \sum_{j=1}^{L} \alpha_{ij}\,v_j. \tag{35}$$

After calculating the weighted sum representation $\mathcal{Z}$, the original input information is fused with the attention output through the residual connection $\widetilde{\mathcal{Z}} = \mathcal{Z} + \mathcal{X}$.

A.13.2. Complex Feedforward Network.

The complex feedforward network consists of two linear mapping layers with a non-linear activation function in between. The feedforward network can be represented as:

$$\mathcal{H} = \sigma\Big(\mathcal{W}_1 \cdot \widetilde{\mathcal{Z}} + \mathcal{B}_1\Big) \tag{36}$$

$$\text{FFN}(\widetilde{\mathcal{Z}}) = \mathcal{W}_2 \cdot \mathcal{H} + \mathcal{B}_2 \tag{37}$$

where $\sigma(\cdot)$ denotes the modReLU activation function. The output is still a complex vector, integrating complex feature information from different positions, thus fulfilling the need for feature integration in the complex domain attention mechanism.

A.13.3. Complex Inner Product.

Assume that for each element we have $q_i = a_i + i\,b_i$, $k_i = c_i + i\,d_i$. Then the conjugate of k_i is given by $\overline{k_i} = c_i - i\,d_i$. Next, we compute the product of each term $q_i \cdot \overline{k_i}$:

$$q_i \cdot \overline{k_i} = (a_i + i\,b_i)(c_i - i\,d_i). \tag{38}$$

Using the distributive law of complex multiplication, this can be expanded as:

$$q_i \cdot \overline{k_i} = a_i c_i - a_i(i\,d_i) + i\,b_i\,c_i - i\,b_i(i\,d_i). \tag{39}$$

Since $i \cdot i = i^2 = -1$, we have:

$$-a_i(i\,d_i) = -i\,a_i\,d_i, \quad -i\,b_i(i\,d_i) = -i^2\,b_i\,d_i = b_i\,d_i. \tag{40}$$

Substituting these, the above equation can be written as:

$$q_i \cdot \overline{k_i} = a_i c_i + b_i d_i + i\,(b_i c_i - a_i d_i). \tag{41}$$

Finally, summing over all elements, we obtain the inner product:

$$\langle q, k \rangle = \sum_{i=1}^{n} q_i \cdot \overline{k_i} = \sum_{i=1}^{n} \left[a_i c_i + b_i d_i + i\,(b_i c_i - a_i d_i)\right]. \tag{42}$$

Furthermore, it can be separated into its real and imaginary parts as:

$$\langle q, k \rangle = \underbrace{\sum_{i=1}^{n}(a_i c_i + b_i d_i)}_{\text{real part}} + i \underbrace{\sum_{i=1}^{n}(b_i c_i - a_i d_i)}_{\text{imaginary part}}. \tag{43}$$

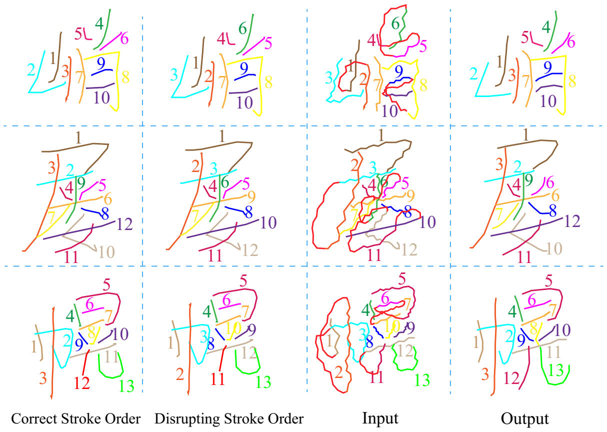


Fig. 11. Impact of incorrect stroke order on stroke optimization. colored numbers indicate the writing order of each stroke.

A.14. Effect of Incorrect Stroke Order

Despite the strict stroke order required for Chinese characters, variations in individual writing habits can result in deviations from the standard sequence. We input characters with a randomized stroke order into the model. As shown in Figure 11, our method effectively adjusts the strokes, restoring high readability in the characters. This finding demonstrates that our approach can accommodate variations in writing errors .

A.15. Different Number of Style Characters

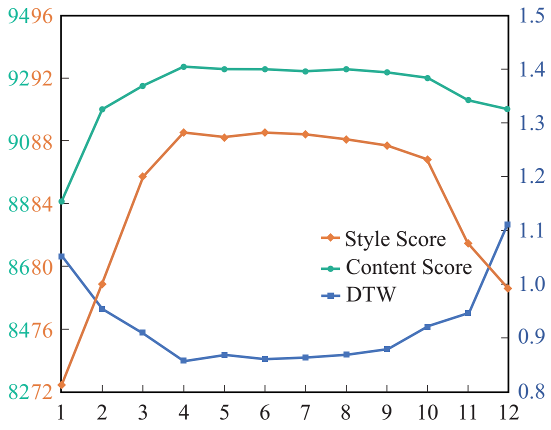


Fig. 12. Impact of the number of style reference characters on model performance.

As shown in Figure 12, this experiment evaluates the impact of the number of style reference characters on model performance. As the number of input style reference characters increases, the style score also improves, indicating that the synthesized handwriting contains richer style information. The model achieves optimal performance when $K = 4$. When the number of style reference characters exceeds 8, the model struggles to capture consistent style features due to the instability of handwriting styles in the dataset. Furthermore, the model performs significantly better in restoring text readability than in learning the writer's unique style.

Modal	Style Score ↑	Content Score ↑	DTW ↓
$\mathcal{N}$	74.35	86.34	1.2453
$\mathcal{F}$	86.32	90.56	0.9156
$\mathcal{D}$	82.45	89.93	0.9767
$\mathcal{C}$	88.35	92.64	0.8745

Table 4. Validation of the Effectiveness of Frequency Domain Filters. $\mathcal{N}$ represents the absence of any filtering. $\mathcal{F}$ represents the static filter, $\mathcal{D}$ represents the dynamic filter, and $\mathcal{C}$ represents the combination of the two filters.

A.16. Receptive Field of the Dynamic Filter

To verify the impact of the local spectral slice size $R_L \times R_D$ on dynamic filter performance, we conducted comparison experiments under five settings: 1×1, 3×3, 7×7, 14×7, and 7×14, while keeping all other parameters unchanged.

When the slice size was 1×1, the dynamic filter reduced to single-point gating, and all three metrics were significantly worse than in the other settings. This indicates that relying only on the current frequency point cannot capture coupling between neighboring spectral points, confirming the importance of local correlation in time and frequency. Expanding the receptive field further led to substantial improvements in all three metrics, demonstrating that a two-dimensional neighborhood can capture key local patterns. We also evaluated two asymmetric window settings. With a window of 7×14, performance exceeded that of 14×7, showing that an excessively long focus along the frequency dimension can introduce clutter and weaken the gating's discriminative power.

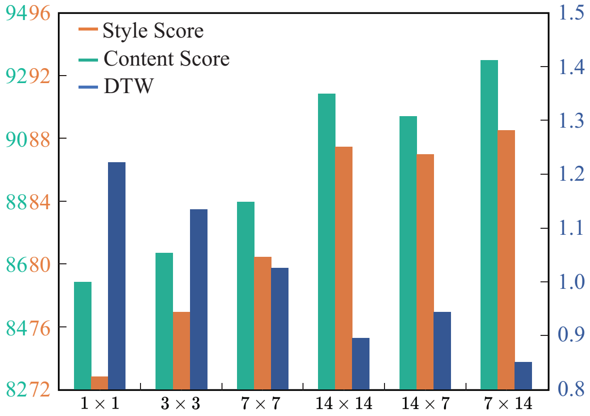


Fig. 13. Model performance across different local spectral window sizes $R_L \times R_D$.

A.17. Effectiveness of Frequency Domain Filters

The validation of the effectiveness of the frequency domain filters is shown in Table 4. The static filter performs better than the dynamic filter when used individually. We used simulated imperfect handwriting as the training set, which contains fixed irregular patterns that are easier for the model to learn. When both filters are combined, the best reconstruction results are achieved.

Modal	Style Score ↑	Content Score ↑	DTW ↓
$\Delta\phi$	78.32	76.73	1.1885
$\Delta\mathcal{A}$	88.35	92.64	0.8745

Table 5. Validation of the Effectiveness of Style Aggregation. $\Delta\mathcal{A}$ represents the perturbation amplitude, and $\Delta\phi$ represents the perturbation phase.

A.18. Effectiveness of Style Aggregation

The validation of the effectiveness of manipulating amplitude components to control the handwriting trajectory style is shown in Table 5. The experimental results confirm our hypothesis: directly perturbing the phase tends to disrupt the original trajectory structure, thereby affecting the character content, while perturbing only the amplitude can adjust the style characteristics without altering the basic structure of the trajectory.

A.19. Failure Case Analysis

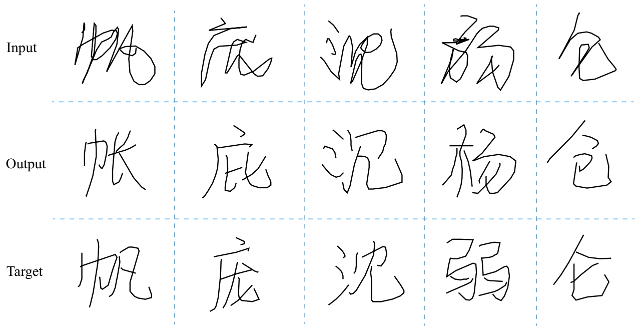


Fig. 14. Erroneous reconstruction of handwritten characters.

In this section, examples of reconstruction failures are presented, as shown in Figure 14. Many Chinese characters exhibit similarities, and characters written in an electronic writing environment, due to the lack of mechanical feedback, tend to further amplify these similarities, making it difficult for the model to accurately capture the subtle differences in strokes between characters. When the input handwritten characters are blurred or incomplete, the model often confuses these details, resulting in erroneous reconstruction outcomes.

A.20. Character Recognition Evaluation

Methods	Input	Acc. ↑	Input	Acc.↑
1D-CNN [6]	$\mathcal{O}$	68.76	$\mathcal{R}$	90.28
DCNN [7]	$\mathcal{O}$	71.67	$\mathcal{R}$	94.55
1D-TCRN [2]	$\mathcal{O}$	75.45	$\mathcal{R}$	95.28
PyGT [8]	$\mathcal{O}$	80.37	$\mathcal{R}$	98.36

Table 6. Handwritten character recognition results. $\mathcal{O}$ represents distorted characters, while $\mathcal{R}$ indicates characters reconstructed using the proposed method.

To further assess the effectiveness of the proposed method, we conducted a quantitative evaluation of recognition accuracy using both distorted characters and reconstructed characters across three state-of-the-art in-air handwriting character recognition methods, indirectly evaluating the quality of reconstructed handwritten strokes. As shown in Tab. 6, the differences in accuracy clearly indicate that our method's reconstructed handwritten strokes significantly enhance the recognition accuracy of in-air handwritten characters.

A.21. Real Handwritten Character Optimization

To evaluate the performance of in-air handwritten character reconstruction, we first used the Apple Vision Pro to capture users' in-air writing trajectory data; concurrently, we employed an Apple Pencil on an iPad to collect a set of neatly written reference characters from the same users. We use the neatly written characters on the iPad as style references to reconstruct the in-air inputs captured by the Apple Vision Pro.

Quantitative results are presented in Tab. 7. The reconstruction performance on real handwriting data is slightly superior to that on the IAHCT-UCAS2018 dataset. This finding demonstrates that the frequency-domain mapping and filtering strategies learned from the IAHCT-UCAS2018 samples can be effectively transferred to real handwriting trajectories, thereby enabling high-quality style and content reconstruction on real data. This validates the effectiveness and robustness of the proposed method in practical scenarios.

Qualitative results are shown in Fig. 15. During reconstruction, stroke start and end points, writing directions, and key bending locations closely align with the reference "ideal" samples. These qualitative results clearly indicate that the proposed method not only restores low-quality handwriting in the simulated environment but also transfers seamlessly to real scenarios, achieving high-quality style and content reconstruction for handwriting trajectories with substantial jitter.

Modal	Style Score ↑	Content Score ↑	DTW ↓
$\mathcal{S}$	88.35	92.64	0.8745
$\mathcal{R}$	89.76	93.21	0.8436

Table 7. Quantitative evaluation of real handwritten character reconstruction. $\mathcal{R}$ denote real handwritten, and $\mathcal{S}$ denote IAHCT-UCAS2018 handwriting.

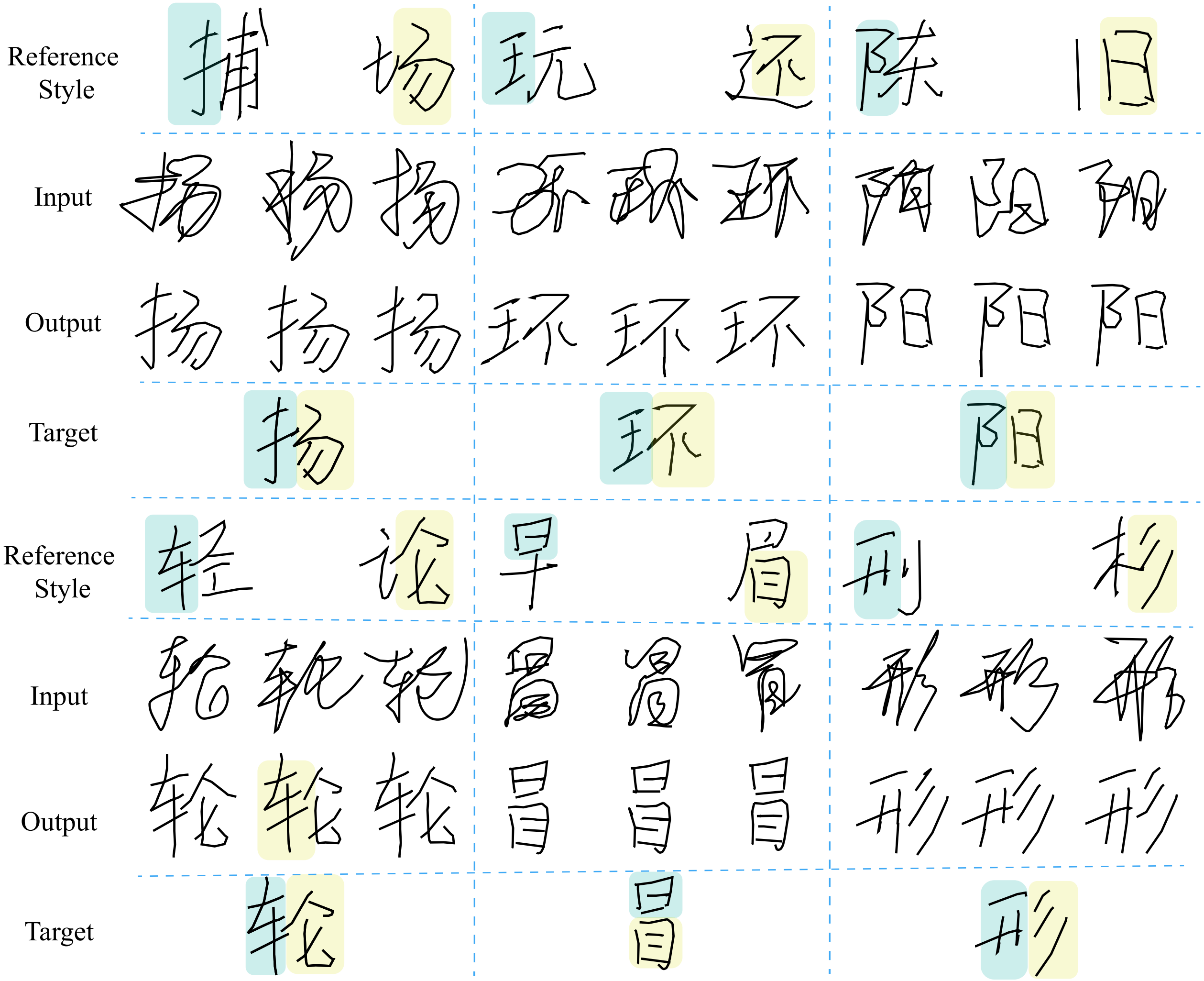


Fig. 15. Qualitative evaluation of real handwritten character reconstruction.

B. REFERENCES

[1] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang, "Casia online and offline chinese handwriting databases," in *2011 international conference on document analysis and recognition*. IEEE, 2011, pp. 37–41.

[2] Ji Gan, Weiqiang Wang, and Ke Lu, "In-air handwritten chinese text recognition with temporal convolutional recurrent network," *Pattern Recognition*, vol. 97, pp. 107025, 2020.

[3] Zhounan Chen, Daihui Yang, Jinglin Liang, Xinwu Liu, Yuyi Wang, Zhenghua Peng, and Shuangping Huang, "Complex handwriting trajectory recovery: Evaluation metrics and algorithm," in *Proceedings of the asian conference on computer vision*, 2022, pp. 1060–1076.

[4] Shusen Tang and Zhouhui Lian, "Write like you: Synthesizing your cursive online chinese handwriting via metric-based meta learning," *Computer Graphics Forum*, vol. 40, no. 2, pp. 141–151, 2021.

[5] Gang Dai, Yifan Zhang, Qingfeng Wang, Qing Du, Zhuliang Yu, Zhuoman Liu, and Shuangping Huang, "Disentangling writer and character styles for handwriting generation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2023, pp. 5977–5986.

[6] Ji Gan, Weiqiang Wang, and Ke Lu, "A new perspective: Recognizing online handwritten chinese characters via 1-dimensional cnn," *Information Sciences*, vol. 478, pp. 375–390, 2019.

[7] Zi-Rui Wang and Jun Du, "Joint architecture and knowledge distillation in cnn for chinese text recognition," *Pattern Recognition*, vol. 111, pp. 107722, 2021.

[8] Ji Gan, Yuyan Chen, Bo Hu, Jiaxu Leng, Weiqiang Wang, and Xinbo Gao, "Characters as graphs: Interpretable handwritten chinese character recognition via pyramid graph transformer," *Pattern Recognition*, vol. 137, pp. 109317, 2023.