WHAT INFORMATION ENABLES CROSS-SCRIPT HANDWRITING LAYOUT SYNTHESIS?

Anonymous authors
Paper under double-blind review

ABSTRACT

Handwriting Layout Synthesis generates the spatial arrangement of handwritten text while preserving writer-specific spacing and layout style, providing the structural foundation for handwriting generation. Current systems achieve strong within-script performance but their ability to generalize to unseen scripts is poorly understood. In this work, we present an investigation of the impact of input token representations on this failure. We conduct a controlled empirical study over the state-of-the-art character-compositional representation as well as script-agnostic, explicit glyph-based, and pretrained visual token representations, introducing the latter three to this task. Conducting our study across three distinct scripts representing the morphosyllabic/logographic, abugida and alphabetic writing systems, we observe that neither character-level vocabulary expansion, nor the removal of script-specific information alone is sufficient to achieve robust cross-script generalization. We find that visual representations substantially outperform character-compositional representations on unseen writing systems, while also achieving modest gains in the in-domain setting. Through representation probing and performance comparisons across representations with increasing information, we show that much of the transfer advantage of pretrained visual representations can be attained using coarse geometric information that is linearly recoverable from the learned embeddings, while visual representations retain a modest in-domain performance advantage. Together, these results provide a principled characterization of how representation choice impacts cross-script robustness in handwriting layout synthesis.

1 INTRODUCTION

Handwriting **L**ayout **S**ynthesis (HLS) generates the spatial arrangement of handwritten text while preserving writer-specific spacing and layout style (Ren et al., 2025). It provides a structural foundation for downstream handwriting synthesis by allowing layout properties such as spacing, alignment, and slant to be modeled separately from character appearance and stroke style (Wang et al., 2026a). Existing handwriting layout models achieve strong performance on scripts represented in their training data (Wang et al., 2026b;a); however, their ability to generalize to unseen writing systems remains largely unexplored.

This problem is particularly relevant because high-quality handwritten datasets remain limited across writing systems (Joshi et al., 2026; Sharma et al., 2024). A model that can transfer layout knowledge across scripts could make better use of available data and reduce the need for script-specific supervision. Cross-script generalization, however, introduces a representation challenge in addition to the modeling challenge: the information provided to the layout predictor may itself be tied to the scripts observed during training.

Existing handwriting layout methods predominantly represent textual input through learned character embeddings. Earlier Chinese handwriting synthesis methods assigned each character a learned embedding (Ren et al., 2025; Wang et al., 2026a), while recent multilingual approaches construct word representations from learned character embeddings Wang et al. (2026b). Such representations are effective within the training distribution, but script shift introduces at least two possible sources of difficulty: First, *characters from an unseen script may not have dedicated learned representations*. Second, *character identity itself may provide limited information that is shared across writing*

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

systems. Cross-script failure may therefore arise not only from the layout prediction model, but also from the information made available through its token representation.

This motivates our central question:

What information supports robust cross-script handwritten layout prediction?

We study this through a controlled empirical comparison in which the layout prediction architecture and training procedure are held fixed while the token representation is systematically varied. Across English, Chinese, and Bengali[1], we compare character-compositional representations, an expanded-vocabulary representation, explicit representations based on character count, token width, height, and ink density, and a frozen pretrained visual representation. This design allows us to examine how script-agnostic information and character identity relate to cross-script performance. Linear probing further examines which geometric properties are encoded in the visual representation and whether they remain recoverable under script shift. Finally, we use ground-truth crop-based reconstruction to assess whether differences in predicted layout translate to differences at the image level, while keeping the underlying handwriting fixed. An overview of the experimental framework and the token representations considered in this study is shown in Figure 1. Our contributions are:

1. *A controlled cross-script representation study.* We provide a like-for-like comparison of character-compositional, script-agnostic, glyph-based, and pretrained visual token representations for cross-script handwriting layout synthesis, including an expanded-vocabulary control that isolates the effect of target-script character coverage.
2. *An information-based analysis of representation transfer.* We construct an information hierarchy of increasingly rich token-level representations, from character count to width, height, and ink density, and measure how their cross-script performance compares with that of a frozen pretrained visual representation. Linear probing complements this analysis by testing which geometric properties are recoverable from the visual embedding within and across scripts.
3. *Downstream validation.* We evaluate the resulting layouts in the handwriting synthesis pipeline using image- and writer-level metrics, assessing whether representation-level differences in layout prediction translate to final handwriting quality.

2 Related Work

Handwriting generation research can generally be categorized into two modalities: *online* and *offline*. Online handwriting records the evolution of pen movements during writing, thereby retaining both temporal dependencies and spatial information associated with individual strokes. It has been investigated extensively with several classes of architectures, including recurrent neural network based methods (Kotani et al., 2020; Zhao et al., 2020; Zhang et al., 2018), diffusion-based models (Luhman & Luhman, 2020; Ren et al., 2024), and transformer-based frameworks (Dai et al., 2023). Offline handwriting, on the other hand, focuses on the rendered visual form of the written content rather than the underlying writing trajectory. It preserves appearance-related properties such as variations in stroke width, curvature, and ink distribution.

2.1 Offline Handwriting Synthesis

Offline handwriting synthesis has progressed from GAN-based approaches (Alonso et al., 2019; Xie et al., 2021; Gan et al., 2022; Kong et al., 2022; Liu et al., 2022) to transformer-based methods such as HWT (Bhunia et al., 2021) and VATr (Pippi et al., 2023a), and more recently to diffusion-based approaches (Wang et al., 2026a;b; Pippi et al., 2025; Dai et al., 2024; Nikolaidou et al., 2023; Ding et al., 2023; Zhu et al., 2023). These methods increasingly support style adaptation from limited reference examples. Many jointly model content, spatial arrangement, and handwriting appearance within the image-generation process (Nikolaidou et al., 2023; Ding et al., 2023; Pippi et al., 2025), while more recent approaches explicitly separate layout generation from subsequent handwriting rendering (Wang et al., 2026a;b).

[1] *Bengali* is also commonly referred to as *Bangla* (Dictionary.com, 2012). The terms are used interchangeably here.

2.2 Handwriting Layout Synthesis

Only a small number of works explicitly treat handwriting layout as a separate generation problem. For online synthesis, Ren et al. (2025) decouple spatial layout from character-stroke generation; for offline synthesis, Wang et al. (2026a) similarly separate layout generation from image synthesis, while Wang et al. (2026b) extend this to multilingual handwriting using word-level conditioning derived from character representations. These works establish layout as a distinct prediction problem, but leave the role of token representation comparatively unexplored. Existing approaches primarily use character-derived representations, such as learned character embeddings or character-compositional representations, which encode identities seen during training but provide less explicit information for characters outside the training vocabulary. We instead systematically examine which token-level information supports layout prediction and cross-script transfer.

2.3 Visual Representations

Visual encodings provide an alternative to character-based representations by representing textual input through its rendered appearance. PIXEL (Rust et al., 2023), for example, represents text directly as images and processes the resulting visual representation with a pretrained vision transformer. Although PIXEL targets language understanding rather than handwriting synthesis, it demonstrates the potential of vocabulary-independent visual representations.

In handwriting generation, Pippi et al. (2023a) similarly exploit visual representations to capture character appearance and style with limited supervision. However, this use of visual information focuses primarily on glyph appearance and rendering rather than token-level spatial layout. In contrast, we investigate whether visual token representations provide useful geometric information for handwriting layout prediction, particularly under cross-script transfer.

3 Approach

3.1 Preliminaries

3.1.1 Data Notation

We consider a dataset of transcribed handwritten text-line images $\mathcal{D} = \{(X^{(j)}, C^{(j)})\}_{j=1}^{N}$, where $X^{(j)}$ denotes a stylized handwritten text-line image and $C^{(j)}$ denotes its corresponding textual content. We represent a token sequence as $C = (c_1, c_2, \ldots, c_n)$, where each c_i denotes a token. We define a token as an arbitrary sequence of one or more characters. Under this definition, a single character constitutes a token in logographic writing systems such as Chinese, whereas a token may correspond to a word in alphabetic writing systems such as English and abugida writing systems such as Bengali.

We represent the layout of a text line using a sequence of token-level bounding boxes, following prior work (Wang et al., 2026a;b). Given a token sequence C, its corresponding layout is denoted by $B = (b_1, b_2, \ldots, b_n)$, where b_i denotes the bounding box associated with token c_i. Each bounding box is parameterized by four quantities, normalised by the width of the input text in pixels.

$$b_i = [w_i, h_i, \Delta x_i, \Delta y_i], \tag{1}$$

representing the token width, token height, horizontal offset from the preceding token, and vertical offset relative to the baseline of the preceding text line, respectively.

3.1.2 Task Analysis

Given a token sequence $C = (c_1, \ldots, c_n)$ and a style reference X_s, the layout model predicts token-level bounding boxes B(Wang et al., 2026b), . We make explicit that each token c_i is first mapped to a token representation $r(c_i)$, giving the representation sequence $R = (r(c_1), \ldots, r(c_n))$. The layout model then predicts

$$p(B \mid R, X_s). \tag{2}$$

We retain the Masked Modeling with Conditional Flow Matching architecture and masking strategy of Wang et al. (2026b) throughout our experiments, varying only the token representation. At each

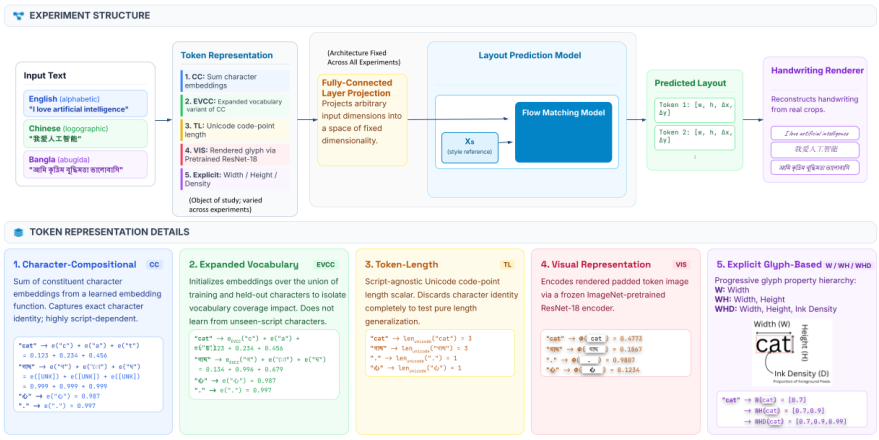

Figure 1: Overview of the experimental framework and token representations. The layout prediction architecture and training procedure are fixed across experiments; only the token representation is varied. The embedding values shown are illustrative and do not reflect the actual values used.

timestep, the model predicts the layouts of masked tokens conditioned on the unmasked token–layout pairs, the current states of the masked tokens, and the style reference. Training uses an $\mathcal{L}_1$ flow-matching loss between predicted and ground-truth velocities for the four bounding-box parameters.

3.2 Token Representations

The layout model does not directly operate on raw token identities, but instead receives a representation of each token. We consider representations with potentially different dimensionalities and project each one into a common d-dimensional space using a fully connected layer. Thus, the representation provided to the layout model is

$$\tilde{r}(c_i) \in \mathbb{R}^d, \qquad \tilde{R} = (\tilde{r}(c_1), \ldots, \tilde{r}(c_n)). \tag{3}$$

Unless otherwise stated, we use R to denote this projected representation sequence when describing the layout model. As summarized in Figure 1, we evaluate representations ranging from character-compositional and token-length encodings to explicit glyph geometry and pretrained visual features.

3.2.1 Character-Compositional Representation

Each token is represented as the sum of its constituent character embeddings obtained from an embedding function e, following (Wang et al., 2026b). For a single-character token, this reduces to the corresponding character embedding. We refer to this as the ***C**haracter-**C**ompositional* (CC) representation, mathematically represented as:

$$r_{\mathrm{CC}}(c_i) = \sum_{j=1}^{|c_i|} e(c_{i,j}) \tag{4}$$

3.2.2 Expanded-Vocabulary Control

During training, e is learned from each training script. To isolate the impact of vocabulary coverage, we take the trained CC model and expand its character-embedding table (e) to (e_{EVCC}) over the union of training and held-out characters, copying the learned entries for training-script characters while leaving held-out entries at their initial values. No further training or parameter updates are performed before proceeding directly to test-set evaluation. We call this the ***E**xpanded **V**ocabulary-**C**haracter **C**ompositional* (EVCC) representation:

$$r_{\mathrm{EVCC}}(c_i) = \sum_{j=1}^{|c_i|} e_{\mathrm{EVCC}}(c_{i,j}) \tag{5}$$

3.2.3 SCRIPT-AGNOSTIC TOKEN-LENGTH REPRESENTATION

To test whether removal of character identity alone is sufficient for cross-script generalisation, we use the Unicode code-point length returned by Python's `len` operation. This script-agnostic ***Token-Length*** (TL) representation discards character identity while retaining only token length. For a token c_i, its representation is defined as

$$r_{\mathrm{TL}}(c_i) = \mathtt{len}_{\mathtt{unicode}}(c_i) \tag{6}$$

3.2.4 FROZEN VISUAL REPRESENTATION

To test whether cross-script transfer can be supported by visual properties that are shared across writing systems, we represent each token using features extracted from its rendered glyphs. We term this the *Visual* (VIS) representation. Each token c_i is rendered as an image using a standard font and encoded with a frozen ImageNet-pretrained ResNet-18 (He et al., 2016). Let $\phi(\cdot)$ denote the pretrained encoder and $I(c_i)$ the rendered token image. The representation is given by

$$r_{\mathrm{VIS}}(c_i) = \phi(I(c_i)) \tag{7}$$

3.2.5 EXPLICIT GLYPH-BASED REPRESENTATIONS

To test whether the performance impact of using visual representations can be explained by lower-dimensional information, we construct representations that progressively encode token visual properties while excluding character identity. For each token c_i, we render its glyphs in a fixed font and extract normalized width w_i, height h_i, and ink density d_i. Width and height are normalized by their 95th-percentile training values, while ink density is the proportion of foreground pixels within the rendered bounding box.

We then consider the following information hierarchy:

$$\begin{aligned} r_{\mathrm{W}} &= f_{\mathrm{W}}(w_i), \\ r_{\mathrm{WH}} &= f_{\mathrm{WH}}(w_i, h_i), \qquad\qquad \mathrm{W} \subset \mathrm{WH} \subset \mathrm{WHD} \\ r_{\mathrm{WHD}} &= f_{\mathrm{WHD}}(w_i, h_i, d_i) \end{aligned} \tag{8}$$

The three representations respectively expose ***Width*** (W), ***Width and Height***, and ***Width, Height and Ink Density*** (WHD). The hierarchy is defined by feature inclusion, not by any assumption that richer representations should yield monotonically better performance.

3.3 REPRESENTATION PROBING

We use linear probes to test whether physical properties of rendered tokens are linearly recoverable from the frozen visual representation. For each token c_i, we independently predict rendered *width*, *height*, *ink density*, and *token length*:

$$\hat{q}_i = W_q r_{\mathrm{VIS}}(c_i) + b_q, \tag{9}$$

where W_q and b_q are learned probe parameters. We report R^2 and Pearson correlation on both training and held-out scripts.

3.4 EVALUATION PROTOCOL

We evaluate representations using layout prediction and ground-truth crop-based handwriting reconstruction. All experiments use the same architecture, training procedure, data protocol, and evaluation procedure; only the token representation varies.

3.4.1 IN-DISTRIBUTION AND CROSS-SCRIPT EVALUATION

We distinguish ***In-Distribution*** (ID) evaluation, where the target script is observed during training, from ***Cross-Script*** (CS) evaluation, where it is held out entirely. We use leave-one-script-out evaluation over English (EN), Chinese (ZH), and Bengali (BN):

$$\mathrm{EN}, \mathrm{ZH} \to \mathrm{BN}, \qquad \mathrm{EN}, \mathrm{BN} \to \mathrm{ZH}, \qquad \mathrm{ZH}, \mathrm{BN} \to \mathrm{EN}. \tag{10}$$

We use EN+ZH→BN as the primary direction and evaluate all representations in this setting. EVCC and VIS are additionally evaluated in all three transfer directions.

Table 1: Summary of writer-disjoint dataset partitions.

DATASET	SYSTEM	SPLIT	WRITERS	FORMS/PAGES	PROPORTION (%)
IAM (English)	Alphabetic	Train	459	1,090	70.8
		Val	98	224	14.6
		Test	100	225	14.6
BN-HTRd (Bengali)	Abugida	Train	104	502	71.9
		Val	21	83	11.9
		Test	23	113	16.2
CASIA HWDB2.0–2.2	Logographic	Train	816	4,076	80.0
		Val	203	1,015	20.0
ICDAR2013 (Chinese)	Logographic	Test	60	300	100.0

3.4.2 Ground-truth Crop-based Reconstruction

To evaluate downstream effects of layout without introducing a handwriting renderer, we extract each token's ground-truth handwritten crop, independently scale it to the predicted width and height, and place it at the predicted spatial location. The resulting line preserves the source handwriting while its spatial arrangement is determined by the predicted layout.

4 Experiments

4.1 Experimental Settings

4.1.1 Datasets

Experiments span three writing systems using four dataset collections: IAM (Marti & Bunke, 2002) for English, BN-HTRd (Rahman et al., 2023) for Bengali, and CASIA HWDB2.0–2.2 (Liu et al., 2011) and ICDAR2013 (Yin et al., 2013) for Chinese.

The English and Bengali data is split into approximately 70/15/15% train/validation/test partitions each, with deviations arising from writer-level splitting. Chinese uses an 80/20% writer-disjoint split of CASIA for training and validation, while ICDAR2013 is held out entirely for testing. All internally partitioned datasets are split at the writer level. The data splits are summarised in Table 1.

4.1.2 Model Selection and Early Stopping

Models are selected exclusively using the ID validation sets, based on mean $\mathcal{L}_1$ loss over the four predicted bounding-box parameters. Early stopping is triggered when no monitored validation set achieves a significant improvement within a predefined patience window, where significance is determined by an empirical noise floor from a baseline training run.

The checkpoint with the lowest mean validation loss across all ID validation sets is selected. Held-out cross-script test sets are evaluated only after model selection is finalized.

4.1.3 Evaluation Metrics

We report mean $\mathcal{L}_1$ loss between predicted and ground-truth token bounding-box parameters for layout prediction. For handwriting reconstruction, we report Kernel Inception Distance (KID) (Binkowski et al., 2018), Fréchet Inception Distance (FID) (Heusel et al., 2017), and Handwriting Distance (HWD) (Pippi et al., 2023b). Lower values indicate better performance for all metrics. Layout metrics are reported separately for each target script, under the ID and CS evaluation conditions defined in Section 3.4.

Table 2: Mean $\mathcal{L}_1$ bounding-box loss ($\times 10^3$). Results for all representations in the primary direction, except TL, are averaged over three random seeds and reported as mean $\pm$ standard deviation. TL is evaluated with a single seed because, as discussed in Section 4.2.2, it induces a structural ambiguity that is independent of random initialization.

(a) Primary representation comparison on EN+ZH→BN.

Representation	In-Domain (ID)	Cross-Script (CS)
CC	$\mathbf{8.042 \pm 0.128}$	$\mathbf{160.934 \pm 14.761}$
EVCC	$\mathbf{8.042 \pm 0.128}$	$\mathbf{22.314 \pm 1.271}$
TL	11.316	17.337
VIS	$\mathbf{7.739 \pm 0.120}$	$\mathbf{14.521 \pm 0.828}$
W	9.378 ± 0.218	$\mathbf{14.201 \pm 0.498}$
WH	8.501 ± 0.147	15.615 ± 0.342
WHD	8.280 ± 0.151	15.405 ± 0.201

(b) EVCC and VIS across transfer directions.

	EN+ZH→BN		EN+BN→ZH		BN+ZH→EN	
Representation	ID	CS	ID	CS	ID	CS
EVCC	8.042 ± 0.128	22.314 ± 1.271	10.304	17.888	8.956	18.960
VIS	7.739 ± 0.120	14.521 ± 0.828	10.142	10.203	8.761	14.231

4.2 Results and Discussion

4.2.1 Impact of Vocabulary Coverage

Table 2a shows the results of the primary representation comparison on EN+ZH→BN. EVCC reduces CS loss from 160.934 to 22.314 relative to CC, with no ID degradation. Because EVCC expands vocabulary coverage while preserving in-domain character representations, this indicates that unseen target characters account for a major portion of the catastrophic CC failure. Figure 3 further shows qualitatively the catastrophic failure of CC and the improvements afforded by the subsequent representations. As shown in Figure 2, this places CC distinctly apart from the other representations in the ID–CS loss space.

4.2.2 Limits of Script-Agnostic Token Length

The TL results in Table 2a reveal that TL improves CS transfer relative to CC but severely degrades ID performance. Because tokens with the same Unicode length receive identical representations, a single-character word and a punctuation mark are indistinguishable, despite potentially very different visual extents, as visible in Figure 3. Thus, script-agnosticism alone is insufficient: *effective transfer is aided by removing character identity but layout-relevant physical information must be retained*.

4.2.3 Visual Representations

Table 2b shows that across all three transfer directions, VIS substantially outperforms EVCC in CS while marginally outperforming EVCC in ID. This shows that its transfer advantage is not explained by vocabulary coverage alone: *rendered visual information provides additional information useful for prediction on unseen scripts*.

4.2.4 Decomposition of the Visual Representation

Table 2a shows all three glyph-based representations achieving lower CS loss than EVCC and remain close to VIS. Notably, W slightly outperforms VIS in CS despite encoding only token width, showing that *a richer representation does not necessarily improve transfer*.

Figure 2: In-domain (ID) versus cross-script (CS) layout loss in the primary EN+ZH → BN setting. Each point denotes one representation; lower values indicate lower loss.

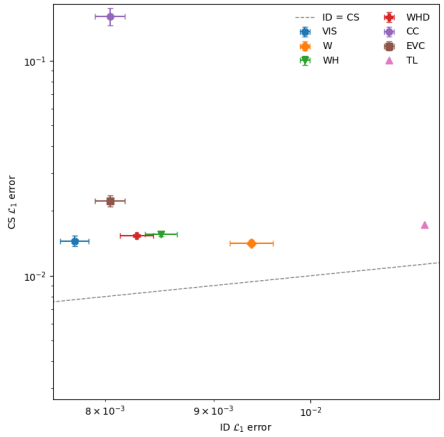

Table 3: Linear-probe performance of frozen VIS representations for recovering token-level properties. Probes are trained on EN+ZH and evaluated on ID samples or the unseen BN script (CS). Values report R^2 with Pearson correlation r in parentheses. Higher is better.

Target property	ID		CS	
Width	**0.989**	**(0.994)**	**0.896**	**(0.969)**
Height	0.689	(0.832)	-0.008	(0.507)
Ink density	0.693	(0.834)	0.044	(0.399)
Character count	0.978	(0.989)	0.709	(0.854)

In contrast, as Figure 2 shows, ID loss decreases monotonically as geometry is enriched from W to WH to WHD, with VIS achieving the lowest loss overall. Thus,*richer glyph-based information improves in-domain prediction but does not translate directly into improved cross-script transfer*.

4.2.5 What Information Is Encoded in the Visual Representation?

Table 3 shows that W is the most robustly transferable property encoded in VIS, with high linear recoverability in both ID ($R^2 = 0.989$) and CS ($R^2 = 0.896$). Height and ink density are substantially recoverable in ID but not under script shift, while character count shows intermediate transferability ($R^2 = 0.709$ in CS). Thus, *VIS may encode multiple geometric properties, but width is the property most consistently preserved across scripts*.

4.3 Downstream Image Quality

The downstream metrics in Table 4 show that VIS improves CS KID, FID, and HWD over EVCC across all three transfer directions, confirming that its *CS layout advantage translates to the renderer-agnostic reconstruction setting*.

Interestingly, *this pattern does not consistently extend to the ID setting*, where EVCC sometimes outperforms VIS. For example, in the BN+ZH→EN direction, the VIS KID, FID and HWD are 0.000643, 25.224, and 0.476, while the EVCC metrics are 0.000596, 24.385, and 0.417, respectively. This indicates that *downstream image metrics are not determined solely by layout $\mathcal{L}_1$ loss and may also reflect reconstruction and metric-specific effects*.

5 Limitations

This study has several limitations. First, the evaluation covers only three writing systems with different tokenization conventions and dataset scales. This reflects the available data, but limits

Table 4: Downstream handwriting reconstruction results across all cross-script transfer directions.

Training Scripts	Held-out Script	Representation	KID ↓	FID ↓	HWD ↓
BN + ZH	EN	VIS	0.000475	21.21	0.868
		EVCC	0.000903	47.82	1.237
BN + EN	ZH	VIS	0.001055	22.97	0.505
		EVCC	0.001847	81.19	1.494
ZH + EN	BN	VIS	0.000552	27.50	0.550
		EVCC	0.001358	63.49	1.287

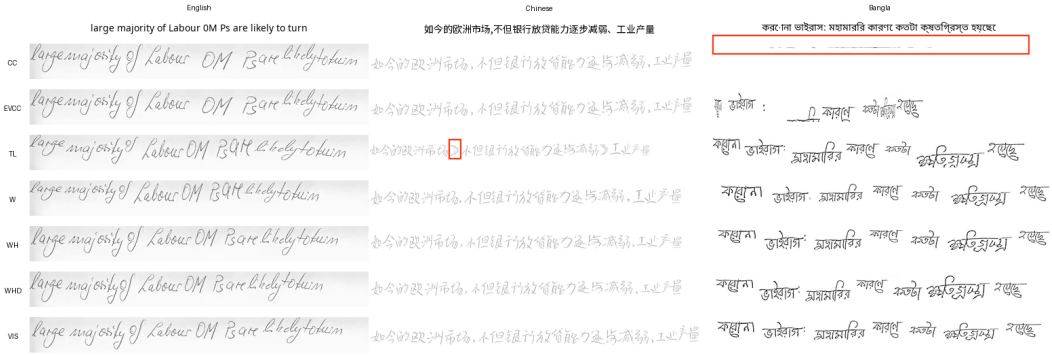

Figure 3: Qualitative comparison of predicted layouts across EN, ZH, and held-out BN. Rows denote token representations; columns denote target scripts, with transcriptions shown above. Red annotations highlight CC's spatial failure on CS and TL's conflation of a comma and a visually larger Chinese character with the same token length.

fully controlled token-level comparisons and potentially contributes to differences across transfer directions. For Chinese, training on CASIA and testing on ICDAR2013 also introduces a dataset shift alongside the intended script shift. Second, VIS uses a fixed font that supports all three scripts, avoiding script-specific font differences as a potential cue for cross-script transfer. However, limited availability of fonts with consistent cross-script coverage means the results should not be interpreted as font-invariant. Finally, downstream evaluation uses renderer-agnostic reconstruction rather than natural handwriting synthesis. Scaling and repositioning handwritten crops can introduce artifacts, meaning image-level metrics may reflect both layout quality and reconstruction effects. A learned handwriting renderer would provide a more realistic end-to-end evaluation but introduce additional factors beyond the representation studied here.

6 Conclusion

We investigated how token representation affects cross-script generalization in handwritten layout synthesis across English, Chinese, and Bengali. Expanding the character vocabulary largely resolves the failure of character-compositional representations on unseen characters, while token length alone discards too much layout-relevant information. Frozen visual representations instead provide substantially stronger cross-script transfer while remaining competitive in-domain, demonstrating that visual token representations provide a robust basis for prediction on unseen scripts.

Geometric analysis shows that token width is strongly recoverable from visual representations across scripts, and an explicit width representation approaches VIS in cross-script performance. However, VIS provides the strongest overall combination of cross-script and in-domain performance in our experiments, indicating that its advantage is not fully explained by any single geometric property. Downstream reconstruction provides further evidence that these representation differences translate to image-level outcomes. Overall, the results suggest that robust cross-script layout prediction depends on transferable physical properties of tokens rather than character identity, motivating more explicitly script-agnostic representations built around such properties.

AI USE STATEMENT

In this work, generative AI tools were used to support manuscript preparation and provide feedback on the experimental methodology. Specifically, AI tools were used for revising portions of the manuscript, improving readability and summarizing portions of existing literature in the literature review phase. They were not used to generate synthetic datasets, propose theoretical models or conceptual frameworks, formulate or prove mathematical claims, propose or refine hypotheses, implement the experimental methods, translate research material, clean or reformat datasets, conduct qualitative or thematic analysis, or create or modify scientific figures or images.

All AI-assisted outputs were reviewed and critically evaluated by the authors before use. Methodological and experimental suggestions were checked against the actual study design and results, while literature-related suggestions were independently verified against the relevant sources. AI-assisted text was revised for factual accuracy, consistency, and appropriate scientific qualification. The authors take full responsibility for the final content, claims, analyses, and other artifacts in this work.

REPRODUCIBILITY STATEMENT

We provide the details necessary to reproduce our experiments across the main paper and supplementary materials. We have ensured the use of only publicly available and accessible datasets throughout our work. The appendices provide additional details of data preprocessing, token rendering, representation construction, model configuration, training and early-stopping procedures, evaluation metrics, and experimental settings. We provide relevant implementation code snippets in the supplementary materials. Upon acceptance, we will release the complete source code and configuration files in a public repository to facilitate reproducibility.

REFERENCES

Eloi Alonso, Bastien Moysset, and Ronaldo Messina. Adversarial generation of handwritten text images conditioned on sequences. In *2019 international conference on document analysis and recognition (ICDAR)*, pp. 481–486. IEEE, 2019.

Ankan Kumar Bhunia, Salman Khan, Hisham Cholakkal, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Mubarak Shah. Handwriting transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1086–1094, October 2021.

Mikolaj Binkowski, Danica J. Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD gans. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1lUOzWCW.

Gang Dai, Yifan Zhang, Qingfeng Wang, Qing Du, Zhuliang Yu, Zhuoman Liu, and Shuangping Huang. Disentangling writer and character styles for handwriting generation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 5977–5986. IEEE, 2023. doi: 10.1109/CVPR52729.2023.00579. URL https://doi.org/10.1109/CVPR52729.2023.00579.

Gang Dai, Yifan Zhang, Quhui Ke, Qiangya Guo, and Shuangping Huang. One-shot diffusion mimicker for handwritten text generation. In *European Conference on Computer Vision*, 2024.

Dictionary.com. Bangla. Dictionary.com, citing Collins English Dictionary—Complete & Unabridged, Digital Edition, 2012. URL https://www.dictionary.com/browse/bangla. Definition: "another name for Bengali".

Haisong Ding, Bozhi Luan, Dongnan Gui, Kai Chen, and Qiang Huo. Improving handwritten ocr withnbsp;training samples generated bynbsp;glyph conditional denoising diffusion probabilistic model. In *Document Analysis and Recognition - ICDAR 2023: 17th International Conference, San José, CA, USA, August 21–26, 2023, Proceedings, Part IV*, pp. 20–37, Berlin, Heidelberg, 2023. Springer-Verlag. ISBN 978-3-031-41684-2. doi: 10.1007/978-3-031-41685-9_2. URL https://doi.org/10.1007/978-3-031-41685-9_2.

Ji Gan, Weiqiang Wang, Jiaxu Leng, and Xinbo Gao. Higan+: Handwriting imitation gan with disentangled representations. *ACM Trans. Graph.*, 42(1), September 2022. ISSN 0730-0301. doi: 10.1145/3550070. URL https://doi.org/10.1145/3550070.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6626–6637, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Prajakta Joshi, Yash Madhukar Patil, Saptarshi Saha, and David Doermann. From pixels to strokes: A survey on generative handwriting. *Int. J. Doc. Anal. Recognit.*, July 2026. URL https://doi.org/10.1007/s10032-026-00609-2.

Yu Kong, Canjie Luo, Weihong Ma, Qi Zhu, Shenggao Zhu, Nicholas Jing Yuan, and Lianwen Jin. Look closer to supervise better: One-shot font generation via component-based discriminator. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13472–13481, 2022. URL https://api.semanticscholar.org/CorpusID:248495910.

Atsunobu Kotani, Stefanie Tellex, and James Tompkin. Generating handwriting via decoupled style descriptors. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII*, pp. 764–780, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58609-6. doi: 10.1007/978-3-030-58610-2_45. URL https://doi.org/10.1007/978-3-030-58610-2_45.

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*, pp. 455–460, 2011. doi: 10.1109/ICDAR.2011.17.

Wei Liu, Fangyue Liu, Fei Din, Qian He, and Zili Yi. Xmp-font: Self-supervised cross-modality pre-training for few-shot font generation. *arXiv preprint arXiv:2204.05084*, 2022.

Troy Luhman and Eric Luhman. Diffusion models for handwriting generation, 2020. URL https://arxiv.org/abs/2011.06704.

U.-V Marti and H. Bunke. The iam-database: An english sentence database for offline handwriting recognition. *International Journal on Document Analysis and Recognition*, 5:39–46, 11 2002. doi: 10.1007/s100320200071.

Konstantina Nikolaidou, George Retsinas, Vincent Christlein, Mathias Seuret, Giorgos Sfikas, Elisa Barney Smith, Hamam Mokayed, and Marcus Liwicki. Wordstylist: styled verbatim handwritten text generation with latent diffusion models. In *International Conference on Document Analysis and Recognition*, pp. 384–401. Springer, 2023.

Vittorio Pippi, Silvia Cascianelli, and Rita Cucchiara. Handwritten Text Generation from Visual Archetypes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023a.

Vittorio Pippi, Fabio Quattrini, , Silvia Cascianelli, and Rita Cucchiara. HWD: A Novel Evaluation Score for Styled Handwritten Text Generation. In *Proceedings of the British Machine Vision Conference*, 2023b.

Vittorio Pippi, Fabio Quattrini, Silvia Cascianelli, Alessio Tonioni, and Rita Cucchiara. Zero-shot styled text image generation, but make it autoregressive. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7910–7919, June 2025.

Md Ataur Rahman, Mitu Paul, Nazifa Tabassum, Riya Pal, Bipon Das, Raisa Tasnim, Md Osman Gony, Fatin Noor, Sheikh Mohammad Jubaer, Mehanaz Chowdhury, Yeasmin Ara Akter, and Mohammad Khairul Islam. Bn-htrd: A benchmark dataset for document level offline bangla handwritten text recognition (htr), 2023.

{Min Si} Ren, {Yan Ming} Zhang, {Qiu Feng} Wang, Fei Yin, and {Cheng Lin} Liu. Diff-writer: A diffusion model-based stylized online handwritten chinese character generator. In Biao Luo, Long Cheng, Zheng-Guang Wu, Hongyi Li, and Chaojie Li (eds.), *Neural Information Processing - 30th International Conference, ICONIP 2023, Proceedings*, Communications in Computer and Information Science, pp. 86–100. Springer Science and Business Media Deutschland GmbH, 2024. ISBN 9789819981403. doi: 10.1007/978-981-99-8141-0_7. Publisher Copyright: © 2024, The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd.; 30th International Conference on Neural Information Processing, ICONIP 2023 ; Conference date: 20-11-2023 Through 23-11-2023.

Minsi Ren, Yan-Ming Zhang, and Yi Chen. Decoupling layout from glyph in online chinese handwriting generation. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=DhHIw9Nbl1.

Phillip Rust, Jonas F. Lotz, Emanuele Bugliarello, Elizabeth Salesky, Miryam de Lhoneux, and Desmond Elliott. Language modelling with pixels. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=FkSp8VW8RjH.

Vandana Sharma, Mohd. Nazim, and Surya Prakash Sharma. A systematic literature review on handwritten text recognition: Approaches, challenges, and future directions. In *International Conference on Innovative Computing Communication (ICICC 2024)*, 2024.

Honglie Wang, Minsi Ren, Yan-Ming Zhang, Fei Yin, and Cheng-Lin Liu. Template-guided cascaded diffusion for stylized handwritten chinese text-line generation. In Xu-Cheng Yin, Dimosthenis Karatzas, and Daniel Lopresti (eds.), *Document Analysis and Recognition – ICDAR 2025*, pp. 149–166, Cham, 2026a. Springer Nature Switzerland. ISBN 978-3-032-04614-7.

Honglie Wang, Yan-Ming Zhang, Wangzi Yao, Fei Yin, and Cheng-Lin Liu. Learning to generate stylized handwritten text via a unified representation of style, content, and noise. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=FBPuLChGNX.

Yangchen Xie, Xinyuan Chen, Li Sun, and Yue Lu. Dg-font: Deformable generative networks for unsupervised font generation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 5130–5140. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.00509. URL https://openaccess.thecvf.com/content/CVPR2021/html/Xie_DG-Font_Deformable_Generative_Networks_for_Unsupervised_Font_Generation_CVPR_2021_paper.html.

Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *2013 12th International Conference on Document Analysis and Recognition*, pp. 1464–1470, 2013. doi: 10.1109/ICDAR.2013.218.

Xu-Yao Zhang, Fei Yin, Yan-Ming Zhang, Cheng-Lin Liu, and Yoshua Bengio. Drawing and Recognizing Chinese Characters with Recurrent Neural Network . *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 40(04):849–862, April 2018. ISSN 1939-3539. doi: 10.1109/TPAMI.2017.2695539. URL https://doi.ieeecomputersociety.org/10.1109/TPAMI.2017.2695539.

Bocheng Zhao, Jianhua Tao, Minghao Yang, Zhengkun Tian, Cunhang Fan, and Ye Bai. Deep imitator: Handwriting calligraphy imitation via deep attention networks. *Pattern Recognition*, 104:107080, 2020. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2019.107080. URL https://www.sciencedirect.com/science/article/pii/S0031320319303814.

Table 5: Layout-generation architecture used in all experiments. All settings are inherited unchanged from the reference implementation corresponding to Wang et al. (2026b).

Component	Setting
Input/output dimension	4 Model dimension
512 Token representation projection	256 Transformer depth
10 Attention heads	8 Head dimension
64 FFN multiplier	4 Positional encoding
Learned sinusoidal Positional convolution	Depthwise 1D convolution Convolution kernel size
31 Normalization	RMSNorm / adaptive RMSNorm Attention normalization
Query/key RMS normalization Objective	Conditional flow matching Prediction head
GEGLU-gated MLP	

Yuanzhi Zhu, Zhaohai Li, Tianwei Wang, Mengchao He, and Cong Yao. Conditional Text Image Generation with Diffusion Models . In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14235–14244, Los Alamitos, CA, USA, June 2023. IEEE Computer Society. doi: 10.1109/CVPR52729.2023.01368. URL https://doi.ieeecomputersociety.org/10.1109/CVPR52729.2023.01368.

A Implementation Details

A.1 Layout Generation Model

We use the layout-generation implementation corresponding to Wang et al. (2026b) from the publicly released TGC-Diff codebase. The repository contains code associated with both Wang et al. (2026a) and Wang et al. (2026b); our experiments use the layout-generation implementation associated with Wang et al. (2026b). We do not modify the model architecture.

The model is a conditional flow-matching estimator operating on four-dimensional token-level layout states. The input box representation has four components, corresponding to horizontal offset, vertical offset, width, and height. The estimator first projects the four-dimensional state and conditioning representation into a hidden dimension of $d = 512$. Token representations are projected to $d_{\text{tok}} = 256$ and concatenated with the state and conditioning representations before projection back to the model dimension.

The estimator uses a 10-layer transformer with 8 attention heads, head dimension 64, and an FFN multiplier of 4. Positional information is incorporated using a depthwise one-dimensional convolutional positional embedding with kernel size 31 and a learned sinusoidal time embedding. RMS normalization, adaptive RMS normalization conditioned on the diffusion time, and query/key RMS normalization are used as in the reference implementation. The prediction head consists of a two-layer gated MLP and outputs the four-dimensional flow field. The architectural settings are summarized in Table 5.

During flow-matching training, the conditional state is denoted by x_0 and an isotropic Gaussian sample by x_1. A time value is sampled as $t = \sigma(z)$, where $z \sim \mathcal{N}(0, 1)$ and σ is the logistic sigmoid. The interpolated state is $x_t = (1 - t)x_0 + tx_1$, and the target vector field is $x_1 - x_0$. The model is trained with an L1 loss on the predicted vector field, evaluated only at positions selected by the prediction mask.

At inference time, the initial state is sampled from a standard Gaussian distribution and the learned vector field is integrated from $t = 1$ to $t = 0$ using the reference implementation's Euler integration procedure.

A.2 Token Definition and Dataset Granularity

A token is defined by the source-dataset annotation unit associated with an ordered handwritten bounding box. We do not perform additional tokenization, linguistic segmentation, or script-specific

retokenization. Each ordered annotated bounding box is therefore treated as one token, and its associated source-dataset string is passed directly to the representation module.

Consequently, the effective token granularity follows the source datasets: English and Bengali annotations are predominantly word-level, whereas Chinese annotations are predominantly character-level. No whitespace-based splitting or subsequent merging of annotations is performed.

For the token-length representation, length is computed directly using Python's `len()` operation on the dataset-provided Unicode string. No grapheme-cluster segmentation is applied.

Strike-through annotations are normalized during construction of the token representation vocabulary. The dedicated strike-through marker is mapped to the fixed visual archetype "XX" for morphology-based representations. Padding is represented separately and is excluded from valid token positions by the sequence masks.

A.3 Token Representations

All token representations are subsequently projected to the common 256-dimensional token representation used by the layout model.

Character-Compositional (CC). For CC, each character is mapped to a learned embedding using the character embedding table, and the embeddings of the characters comprising a token are summed. Padding uses index zero and therefore contributes zero to the summed representation.

Expanded-Vocabulary Character-Compositional (EVCC). EVCC uses the same character-compositional construction while expanding the character vocabulary to include the held-out target characters. Character entries observed during training retain their learned embeddings, whereas newly introduced held-out characters are assigned entries in the expanded embedding table without target-script training updates. This isolates the effect of vocabulary coverage from the compositional representation itself.

Token-Length (TL). TL represents each token by a single scalar equal to its Unicode string length, computed using Python's `len()` function. The scalar is projected to the 256-dimensional token representation space using a learned linear layer.

Visual (VIS). VIS is constructed by rendering each token using a fixed font and extracting a frozen ImageNet-pretrained ResNet-18 representation. Non-Hanzi/non-Korean tokens are rendered using `GoNotoCurrent-Regular.ttf`, while simplified Chinese tokens are rendered using `NotoSansSC-Regular.ttf`. Both fonts are rendered at font size 64. These fonts were chosen because of their uniform style across an extensive character set.

For each token, the first supplied font capable of rendering all required glyphs is selected. The rendered alpha channel is converted to a grayscale image and inverted so that ink corresponds to lower intensity values. The image is then passed to a fixed-size preprocessing procedure with maximum width 384 and height 64. Aspect ratio is preserved; images are never upscaled, and images exceeding the target dimensions are downsampled using Lanczos interpolation. The resulting image is centered on a white 384×64 canvas.

The grayscale image is replicated across three channels and scaled to $[0, 1]$. ImageNet channel-wise normalization is then applied using mean $(0.485, 0.456, 0.406)$ and standard deviation $(0.229, 0.224, 0.225)$. The resulting images are passed through `torchvision` ResNet-18 initialized with ImageNet-1K weights (`IMAGENET1K_V1`). The final classification layer is replaced by the identity mapping, yielding a 512-dimensional feature vector. All ResNet parameters are frozen. The resulting 512-dimensional features are precomputed and stored for each unique token. During layout-model training, these features are projected from 512 to 256 dimensions by a learned linear layer.

Width (W). W represents a rendered token by its W normalized by the fixed maximum rendering width:

$$w_{\text{rep}} = \frac{w_{\text{render}}}{384}.$$

Width and Height (WH). WH augments the width representation with the rendered height normalized by the fixed font size:

$$\mathbf{r}_{\text{WH}} = \left[\frac{w_{\text{render}}}{384}, \frac{h_{\text{render}}}{64} \right].$$

Width, Height, and Ink Density (WHD). WHD further includes the fraction of rendered pixels whose grayscale intensity is below 255:

$$\mathbf{r}_{\text{WHD}} = \left[\frac{w_{\text{render}}}{384}, \frac{h_{\text{render}}}{64}, \frac{1}{WH} \sum_{i,j} \mathbb{1}[I_{ij} < 255] \right].$$

Thus, the density component measures the fraction of the rendered image occupied by non-white pixels.

The W, WH, and WHD representations are passed through learned linear projections to the common 256-dimensional token representation space.

A.4 Rendering and Visual Feature Extraction

Token images are generated deterministically from the configured fonts. For each token, the rendering procedure tests the supplied fonts in order and selects the first font for which all required glyph metrics are available. The rendered surface's alpha channel is extracted and transposed into image coordinates. For morphology-based representations, the alpha image is inverted to obtain the grayscale image used to calculate rendered dimensions and ink density.

For VIS, the rendered image is passed through the fixed 384×64 padding and alignment procedure described above before ImageNet normalization and ResNet-18 feature extraction. Token-level visual features are computed once for every unique token and cached, so repeated occurrences of the same token use the same visual representation.

A.5 Training

We optimize the layout model using AdamW with an initial learning rate of 10^{-4}, weight decay 10^{-4}, and a training batch size of 160. Training is capped at 120,000 iterations. Validation and checkpoints are performed every 10 epochs.

Early stopping uses a patience window of 20,000 iterations measured from the most recent statistically significant validation-loss improvement. The significance threshold is 4×10^{-4}; this threshold is used only for validation-loss improvement detection and is not the Gaussian noise used by the flow-matching objective.

During training, normalized line-location inputs receive independent Gaussian perturbations with standard deviation 0.001. The token masking procedure randomly selects either a contiguous token segment or independently masks tokens, while ensuring that at least one token is masked. At validation and test time, the dataset-provided target mask is used instead of stochastic training-time masking.

All reported results are obtained using random seeds 1001, 1729, and 2027. Unless otherwise stated, all representation variants use the same layout-generation architecture, optimization settings, data splits, and training procedure.

Table 6: Downstream handwriting quality evaluation results. Each row corresponds to a representation and training-script pair. KID, FID, and HWD are reported separately for Bengali (BN), Chinese (ZH), and English (EN).

Representation	Training scripts	KID-std			FID			HWD		
		BN	CN	EN	BN	CN	EN	BN	CN	EN
VIS	BN + ZH	0.000643	0.000785	0.000475	25.2238	29.7918	21.2057	0.4755	0.8047	0.8676
EVCC	BN + ZH	0.000596	0.000627	0.000903	24.3849	26.6715	47.8161	0.4170	0.6984	1.2370
VIS	BN + EN	0.000556	0.001055	0.000719	23.6518	22.9655	27.5608	0.4303	0.5048	0.8845
EVCC	BN + EN	0.000548	0.001847	0.000475	24.7386	81.1902	18.2935	0.4773	1.4939	0.7231
VIS	ZH + EN	0.000552	0.000642	0.000368	27.4982	22.1895	14.2733	0.5504	0.7162	0.6274
W	ZH + EN	0.000606	0.000895	0.000618	27.1147	36.7522	27.4387	0.6563	0.8004	0.9764
WH	ZH + EN	0.000630	0.000780	0.000665	27.4466	34.5940	24.0188	0.6023	0.7679	0.8259
WHD	ZH + EN	0.000628	0.000979	0.000474	27.4753	37.9693	16.2322	0.5610	0.8367	0.6680
CC	ZH + EN	0.003964	0.000622	0.000547	272.5047	23.6560	20.5444	2.6020	0.6991	0.7624
EVCC	ZH + EN	0.001358	0.000622	0.000547	63.4942	23.6560	20.5444	1.2866	0.6991	0.7624
TL	ZH + EN	0.000511	0.001040	0.000661	26.5955	56.0968	29.4227	0.5980	0.9179	0.9336

A.6 Layout Reconstruction for Downstream Evaluation

For downstream image-based evaluation, predicted layouts are converted to pixel coordinates using the input image width. Predicted normalized coordinates are multiplied by the image width and rounded to integer coordinates. The horizontal and vertical offsets are shifted so that their minima are zero. Tokens are then placed sequentially according to the predicted horizontal offsets and the cumulative widths of preceding tokens.

To reconstruct a handwritten paragraph, we use the corresponding ground-truth handwritten crop for each token rather than a learned image renderer. Each ground-truth token crop is resized to the predicted token width and height using Lanczos interpolation. A local background estimate is obtained by applying a 31×31 Gaussian blur to the resized grayscale crop. Assuming black ink over the local paper background, an opacity map is estimated as

$$\alpha = \mathrm{clip}\left(1 - \frac{I}{\max(B, 1)}, 0, 1\right),$$

where I is the resized grayscale crop and B is its locally estimated background. The crop is represented as black RGB pixels with this recovered alpha channel and alpha-composited at the predicted location on the paragraph background.

This procedure changes the spatial arrangement and predicted dimensions of the ground-truth handwritten token crops while preserving their handwritten appearance. Consequently, the downstream reconstruction metrics evaluate the consequences of layout prediction rather than the quality of a separately trained handwriting renderer.

B Additional Results

B.1 Comprehensive Downstream Handwriting Quality Metrics

Table 6 details the downstream performance of each representation under each transfer direction.

B.2 Comprehensive Linear Probe Results

Table 7 records the results of the representation probes across multiple targets.

B.3 Experimental Configuration

Table 8 shows the specific step at which the best performing model's training was stopped for each representation across each training script across each random seed.

Table 7: Probe analysis of the information encoded by each token representation for the Chinese+English training configuration. Linear probes predict token-level width, height, ink density, and character count from the learned token representations. Results are reported as mean ± standard deviation over three runs ($n = 3$). R^2 and Pearson correlation are reported separately for in-vocabulary (IV) and out-of-vocabulary (CS) tokens.

Representation	Split	Density		Height		Num. chars		Width	
		R^2	r	R^2	r	R^2	r	R^2	r
VIS	IV	0.6930 ± 0.0048	0.8337 ± 0.0025	0.6888 ± 0.0058	0.8318 ± 0.0032	0.9779 ± 0.0003	0.9889 ± 0.0001	0.9887 ± 0.0001	0.9943 ± 0.0001
VIS	CS	0.0435 ± 0.1167	0.3986 ± 0.0707	-0.0084 ± 0.0412	0.5075 ± 0.0348	0.7094 ± 0.0190	0.8540 ± 0.0051	0.8964 ± 0.0120	0.9689 ± 0.0024
W	IV	0.3570 ± 0.0000	0.5992 ± 0.0000	-0.0012 ± 0.0000	0.0372 ± 0.0006	0.9533 ± 0.0000	0.9765 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
W	CS	0.2305 ± 0.0000	0.4872 ± 0.0000	-0.2649 ± 0.0000	0.4851 ± 0.0001	0.8682 ± 0.0000	0.9331 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WH	IV	0.4678 ± 0.0000	0.6845 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	0.9536 ± 0.0000	0.9767 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WH	CS	0.4624 ± 0.0000	0.7757 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	0.8578 ± 0.0000	0.9280 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WHD	IV	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	0.9582 ± 0.0000	0.9790 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
WHD	CS	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000	0.8481 ± 0.0000	0.9248 ± 0.0000	1.0000 ± 0.0000	1.0000 ± 0.0000
CC	IV	0.4736 ± 0.0242	0.6950 ± 0.0161	0.4704 ± 0.0074	0.6893 ± 0.0056	0.9889 ± 0.0005	0.9945 ± 0.0002	0.9654 ± 0.0004	0.9826 ± 0.0002
CC	CS	-66.7792 ± 9.5102	-0.5068 ± 0.0016	-17.3557 ± 14.0972	-0.6016 ± 0.0028	-2.2042 ± 3.4498	0.4752 ± 0.8532	-8.0142 ± 10.2761	0.1944 ± 0.8825
EVCC	IV	0.4736 ± 0.0242	0.6950 ± 0.0161	0.4704 ± 0.0074	0.6893 ± 0.0056	0.9889 ± 0.0005	0.9945 ± 0.0002	0.9654 ± 0.0004	0.9826 ± 0.0002
EVCC	CS	-4.8354 ± 1.6828	-0.0405 ± 0.1221	-1.2021 ± 0.9872	0.0282 ± 0.3067	-4.9695 ± 2.0343	-0.0133 ± 0.3982	-4.8390 ± 2.2632	0.0195 ± 0.3294
TL	IV	0.3942	0.6293	−0.0021	0.0189	1.0000	1.0000	0.9534	0.9765
TL	CS	0.2738	0.5288	−0.2749	0.5974	1.0000	1.0000	0.8681	0.9331

Table 8: Experimental configurations and training checkpoints. The reported epoch and training step correspond to the selected checkpoint for each run.

Representation	Training scripts	Seed	Epoch	Step
VIS	BN + ZH	1001	779	62399
VIS	BN + EN	1001	2329	62909
VIS	ZH + EN	1001	1319	104279
VIS	ZH + EN	1729	1119	88479
VIS	ZH + EN	2027	929	73469
WHD	ZH + EN	1001	999	78999
WHD	ZH + EN	1729	1119	88479
WHD	ZH + EN	2027	869	68729
WH	ZH + EN	1001	1089	86109
WH	ZH + EN	1729	1369	108229
WH	ZH + EN	2027	869	68729
W	ZH + EN	1001	859	67939
W	ZH + EN	1729	889	70309
W	ZH + EN	2027	869	68729
CC	ZH + EN	1001	879	69519
CC	ZH + EN	1729	1069	84529
CC	ZH + EN	2027	979	77419
EVCC	BN + ZH	1001	689	55199
EVCC	BN + EN	1001	1759	47519
EVCC	ZH + EN	1001	879	69519
EVCC	ZH + EN	1729	1069	84529
EVCC	ZH + EN	2027	979	77419
TL	ZH + EN	1001	419	33179

B.4 In-Vocabulary and Out-of-Vocabulary Evaluation

Table 9 shows the IV and CS testing losses for each individual seed across the primary transfer direction.

Table 9: In-vocabulary (IV) and out-of-vocabulary (CS) testing losses for each experimental run.

Representation	Seed	IV loss	CS loss
VIS	1001	0.007601	0.014182
VIS	1729	0.007822	0.013916
VIS	2027	0.007793	0.015464
WHD	1001	0.008285	0.015444
WHD	1729	0.008126	0.015583
WHD	2027	0.008428	0.015187
WH	1001	0.008419	0.015685
WH	1729	0.008412	0.015917
WH	2027	0.008671	0.015244
W	1001	0.009382	0.013706
W	1729	0.009158	0.014703
W	2027	0.009594	0.014193
CC	1001	0.007911	0.166939
CC	1729	0.008047	0.144116
CC	2027	0.008166	0.171746
EVCC	1001	0.008956	0.018960
EVCC	1729	0.008047	0.021492
EVCC	2027	0.008166	0.023779
TL	1001	0.011316	0.017337