# CALLIMASTER: MASTERING PAGE-LEVEL CHINESE CALLIGRAPHY VIA LAYOUT-GUIDED SPATIAL PLANNING

**Anonymous authors**
Paper under double-blind review

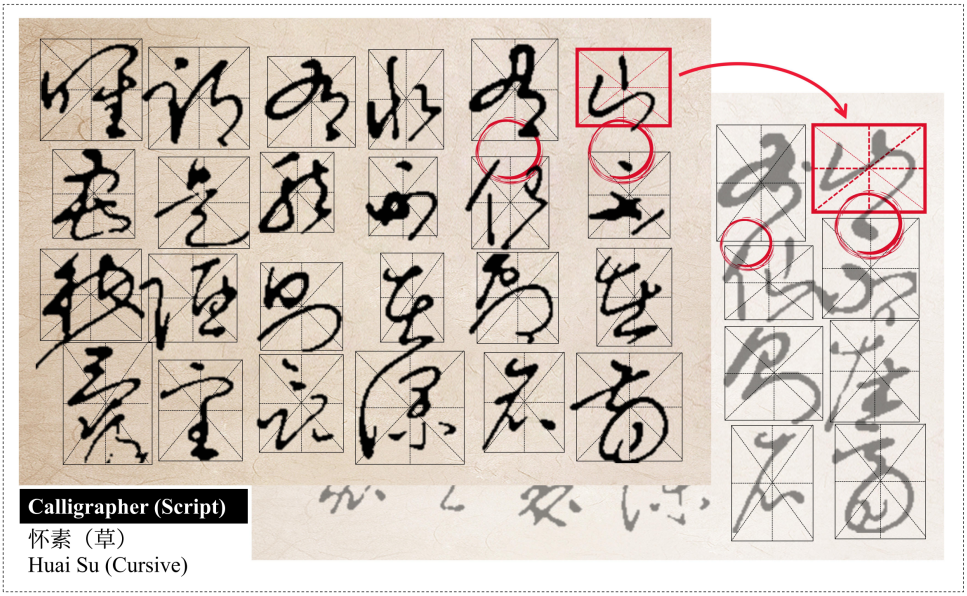



Figure 1: **Page-level generation and editing.** Left: a page generated from target text and style. Right: edited character boxes (□) guide regeneration of the page, including visible inter-character connections (◯).

## ABSTRACT

Chinese calligraphy requires control of placement, style, and inter-character strokes: style shapes spacing and scale, while connections depend on neighboring characters. Character-level methods miss this context; page-level and general-purpose models struggle with spatial coherence or brushwork fidelity. Thus, we present **CalliMaster**, which plans before writing: given standard-font text and a style description, it predicts a page layout, then renders calligraphy conditioned on it. One multimodal diffusion transformer performs both stages. Rasterized character boxes and calligraphy pages share a visual latent space, while independent noise schedules and an attention mask support layout-to-image generation. The boxes enable editing: after users revise the box layout, CalliMaster regenerates the page with adapted brushwork and visible connections. With predicted layouts, it achieves the highest reference-based similarity scores among baselines and a 4.68/5 overall rating in the reported user study. Under a matched training budget, it improves image and layout metrics over separate stage models. It also supports restoration under simulated damage and exploratory reconstruction-based authenticity analysis across multiple noise scales.

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.

# 1 INTRODUCTION

A Chinese calligraphy page is more than a collection of well-formed characters. Its visual rhythm depends on character scale, spacing, and surrounding negative space; its brushwork depends on stroke texture and connections that carry movement across characters (Chen, 2017; Qiu, 2022). At the page scale, the writer anticipates where each character will sit and how the brush will move into the next. Within a character, changes in pressure and ink deposition shape individual strokes. These decisions interact: enlarging or shifting one character changes the space available to its neighbors and the path of the connecting stroke. The empty regions are therefore part of the composition, not unused background between independent glyphs. Unlike printed text, these choices also vary with the writer's style. A generator must therefore coordinate page composition with the rendering of each character. Focusing on one scale alone risks producing either accurate glyphs in an awkward arrangement or a coherent page with generic brushwork.

Existing methods address only parts of this coupled problem. Isolated-character synthesis (Zhang et al., 2024; Yang et al., 2024; Xie et al., 2021) can produce accurate glyphs but does not model page composition or strokes that cross character boundaries. UniCalli (Xu et al., 2026) generates columns jointly, yet extending this to style-dependent full-page layouts remains challenging. CalliPaint (Liao et al., 2023) supports inpainting, and CalliffusionV2 (Liao et al., 2024) provides multimodal control, but neither explicitly plans an editable page layout before rendering. Without such an intermediate plan, changing the size or position of one character cannot directly guide the placement and rendering of its neighbors. This matters especially in running and cursive scripts, where character proportions and connections vary with style. General-purpose image generation models (Labs et al., 2025; OpenAI, 2025) can compose complete images but often produce incorrect characters or generic, font-like strokes. These limitations point to the need for a page-level model that makes spatial planning explicit and uses the resulting plan to guide brushwork.

To address this gap, we introduce **CalliMaster** (Fig. 1), which follows the practice of "planning before writing". Given target text rendered in a standard font and a style description, it first predicts a style-conditioned page layout of character bounding boxes. This plan sets position and scale across the page, including the negative space between glyphs. The second stage uses flow matching to render the page under that layout, allowing glyph shapes and connecting strokes to respond to neighboring characters. Both stages run in a single multimodal diffusion transformer. The boxes are rasterized and encoded through the same visual pathway as the calligraphy image, so layout and image tokens occupy a shared visual space without a separate coordinate prediction head. Style enters through text conditioning, allowing the plan itself to vary with the intended calligrapher rather than applying one fixed grid to every page. Diffusion Forcing-inspired independent noise schedules (Chen et al., 2024) and a causal attention mask support the Text $\rightarrow$ Layout $\rightarrow$ Image order during inference. The mask prevents layout tokens from reading image tokens, while image tokens can attend to the completed layout.

Because the layout is explicit, it also becomes an editing interface. Users can move or resize boxes, or insert or delete them with matching text and font-image updates, and supply the revised layout to the rendering stage. In practice, the predicted layout is decoded into rectangles that can be manipulated and then rasterized again for the model. The model regenerates the page rather than warping its original pixels, adapting brushwork, inter-character connections, and the use of empty space to the new geometry. For example, resizing a title character can change nearby stroke paths even when the text remains fixed. The same network is used for restoration under simulated damage. Its reconstruction error also provides an exploratory signal for authenticity analysis; we do not treat this as a validated authentication test.

Our contributions are threefold: **(1) Unified page-level generation.** CalliMaster performs layout planning and layout-guided rendering in one diffusion transformer with a shared visual representation. Under a matched training budget, it improves image and layout metrics over separately trained stage models. It also leads evaluated baselines on reference-based similarity and receives the highest overall rating in our user study. **(2) Layout-guided editing.** Character boxes act as editable geometric prompts, enabling the model to regenerate brushwork and connections after layout changes. [1] **(3) Evaluation and extensions.** We report Layout EMD for box geometry, Rhythm EMD for variation

[1] Interactive examples are provided in the supplementary material (`project_page/index.html`).

in spacing and scale, and Layout–Ink Align for agreement between planned boxes and rendered ink. We also demonstrate restoration under simulated damage and exploratory authenticity analysis.

## 2 Related Work

### 2.1 From Isolated Characters to Page-Level Calligraphy

Early GAN-based (Wu et al., 2020; Xie et al., 2021) and recent diffusion-based methods (Zhang et al., 2024; He et al., 2024; Hu et al., 2026) focus on isolated characters, so they do not model page rhythm or inter-character stroke continuity. CalliPaint (Liao et al., 2023) uses diffusion-based inpainting, while CalliffusionV2 (Liao et al., 2024) supports text- and image-conditioned character generation; neither predicts an editable page layout as a separate planning stage. Closer to us, DiffInk (Pan et al., 2026) and UniCalli (Xu et al., 2026) model an entire line or column jointly, avoiding the stitching artifacts of character-wise pipelines; yet neither treats layout as a prerequisite plan: DiffInk keeps it implicit in pen trajectories without brush texture, while UniCalli generates boxes *simultaneously* with the pixels as a parallel co-objective. CalliMaster instead follows Text $\rightarrow$ Layout $\rightarrow$ Image at the page level, with boxes as editable geometric priors for brushwork generation.

### 2.2 Sequence Modeling in Diffusion Frameworks

Diffusion models (Ho et al., 2020; Song et al., 2020) dominate high-fidelity image synthesis, yet standard architectures lack the precise spatial planning that sequence generation requires. MM-DiT (Esser et al., 2024; Labs et al., 2025) employs bidirectional attention over multimodal sequences, while Diffusion Forcing (Chen et al., 2024) bridges next-token prediction and diffusion by assigning independent, per-token noise levels, which permits "causal" generation—planning future tokens from past ones—without losing global context. CalliMaster uses independent noise levels to train planning and rendering within one network. Its staged inference procedure finalizes the spatial plan before pixel synthesis.

### 2.3 Layout-Guided Editing and Restoration

Scene text editors such as AnyText (Tuo et al., 2024) and TextDiffuser-2 (Chen et al., 2023) are not designed for calligraphic stroke connections after layout edits; restoration models such as DiffACR (Li et al., 2024) focus on filling missing image regions. We use character boxes as geometric prompts to regenerate strokes and their visible spacing under an edited layout.

## 3 Method

We present CalliMaster (shown in Fig. 2), a unified framework for Chinese calligraphy generation and editing. The core idea is to decompose the complex page-level synthesis task into two tractable sub-problems: **spatial planning** and **content synthesis**. We achieve this by extending a rectified flow transformer with a Diffusion Forcing-inspired multi-timestep architecture. In this section, we describe the overall formulation (Sec. 3.1), the model architecture (Sec. 3.2), and the multi-objective training strategy (Sec. 3.3). Finally, we detail the inference procedure and various downstream applications (Sec. 3.4).

### 3.1 Problem Formulation

Given a text prompt $\mathbf{p}$ specifying the characters to render and a style description $\mathbf{s}$ (e.g., calligraphic script, author), we also use an auxiliary reference condition image $\mathbf{c}$. It provides foundational structural font priors by depicting standard-font glyphs arranged in an initial page layout. Our goal is to generate a calligraphy image $\mathbf{x}$ that (i) faithfully renders the specified characters, (ii) captures the target artistic style, and (iii) preserves global spatial coherence, including the rhythm of density and void that defines a calligraphic composition.

We adopt a *rectified flow* formulation (Liu et al., 2022; Esser et al., 2024). With $\mathbf{x}_0$ the clean data and $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ Gaussian noise, the forward interpolation at noise level $t \in [0, 1]$ is:

$$\mathbf{x}_t = (1 - t)\,\mathbf{x}_0 + t\,\boldsymbol{\epsilon}, \tag{1}$$

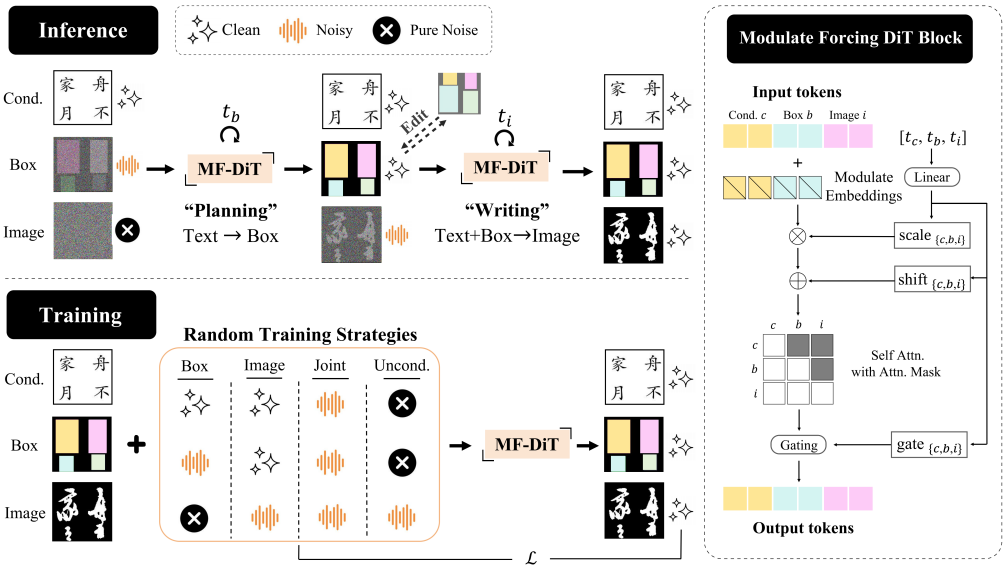



Figure 2: **Overview of CalliMaster.** Inference plans a box layout, then renders the page. Training samples one of four regimes; the Box and Image columns are S1 and S2 (Sec. 3.3). MF-DiT is our MM-DiT block with modality-aware AdaLN over independent timesteps ($t_c$, $t_b$, $t_i$), a structured attention mask, and modality ("Modulate") embeddings. Text and style tokens, omitted for clarity, join the sequence; their global embedding enters AdaLN (Eq. (6)).

and the model is trained to predict the velocity field $\mathbf{v} = \boldsymbol{\epsilon} - \mathbf{x}_0$.

**From Single to Multi-Timestep.** Unlike standard flow matching where a single timestep $t$ governs the entire input, we introduce *independent noise schedules* for three token groups (Image, Box, and Condition) inspired by Diffusion Forcing (Chen et al., 2024). We define a timestep triplet $(t_{\text{img}}, t_{\text{box}}, t_{\text{cond}})$, where each component controls the noise level of its respective modality:

$$\mathbf{x}_t^{\text{img}} = (1 - t_{\text{img}})\,\mathbf{x}_0 + t_{\text{img}}\,\boldsymbol{\epsilon}_{\text{img}}, \tag{2}$$

$$\mathbf{b}_t^{\text{box}} = (1 - t_{\text{box}})\,\mathbf{b}_0 + t_{\text{box}}\,\boldsymbol{\epsilon}_{\text{box}}, \tag{3}$$

$$\mathbf{c}_t^{\text{cond}} = (1 - t_{\text{cond}})\,\mathbf{c}_0 + t_{\text{cond}}\,\boldsymbol{\epsilon}_{\text{cond}}, \tag{4}$$

where $\mathbf{b}_0$ and $\mathbf{c}_0$ are the clean box layout and condition image latents, respectively.

### 3.2 Model Architecture

CalliMaster uses an MM-DiT backbone (Peebles & Xie, 2023; Labs et al., 2025) with a unified sequence of text and visual tokens.

**Token Sequence Construction.** A frozen VAE encodes each visual stream, which is patchified into $N$ tokens using $p \times p$ patches. The composite sequence is

$$\mathbf{Z} = [\underbrace{\mathbf{z}_{\text{txt}}^{1:L}}_{\text{text}}, \underbrace{\mathbf{z}_{\text{img}}^{1:N}}_{\text{image}}, \underbrace{\mathbf{z}_{\text{box}}^{1:N}}_{\text{box layout}}, \underbrace{\mathbf{z}_{\text{cond}}^{1:N}}_{\text{condition}}], \tag{5}$$

where $L$ denotes the text sequence length. While text tokens are projected linearly from a text encoder embedding, the three visual streams $(\text{img}, \text{box}, \text{cond})$ share a common input projection layer initialized with weights $W_{\text{in}}$. To disambiguate these streams within the shared latent space, we introduce learnable **modality embeddings** $\mathbf{e}_m$ for $m \in \{\text{img}, \text{box}, \text{cond}\}$, such that the input to the transformer is $\mathbf{z} = W_{\text{in}}\,\mathbf{p}_{\text{patch}} + \mathbf{e}_m$.

To further encode spatial correspondence and modality identity, we extend 3D Rotary Position Embeddings (RoPE) (Su et al., 2024) to jointly represent the modality index $m$ and 2D coordinates

$(y, x)$, ensuring the model distinguishes overlapping visual streams while maintaining their spatial alignment.

Character boxes are rasterized with random colors and encoded through the shared visual pathway, requiring no layout-specific encoder or coordinate prediction head. Editing coordinates are external metadata (Sec. B.4).

**Modality-Aware Adaptive Layer Normalization.** A core contribution of our architecture is the decoupling of noise schedules across modalities. We implement this via *modality-aware AdaLN* (Peebles & Xie, 2023). Each modality $m$ is assigned an independent timestep $t_m$, which is mapped to a conditioning vector $\mathbf{h}_m$ via a shared timestep embedder:

$$\mathbf{h}_m = \mathrm{MLP}_{\mathrm{time}}(\gamma(t_m)) + \mathrm{MLP}_{\mathrm{vec}}(\mathbf{y}_{\mathrm{txt}}), \tag{6}$$

where $\gamma$ is a sinusoidal encoding. The frozen text encoder processes the concatenated content prompt $\mathbf{p}$ and style description $\mathbf{s}$, supplying both attention tokens and the global embedding $\mathbf{y}_{\mathrm{txt}}$. Each block derives modality-specific scale, shift, and gate parameters from $\mathbf{h}_m$:

$$\hat{\mathbf{z}}^{(i)} = \left(1 + \alpha_m^{(i)}\right) \cdot \mathrm{LN}(\mathbf{z}^{(i)}) + \beta_m^{(i)}, \quad m = \mathrm{mod}(i), \tag{7}$$

where $\mathrm{mod}(i)$ identifies the modality of the $i$-th token. This mechanism allows the network to process different modalities at varying noise levels simultaneously, such as denoising the box layout while keeping the image tokens fully noisy.

**Causal Attention Mask.** To enforce a coarse-to-fine generation hierarchy, we impose a structured attention mask $\mathbf{A}$. We define a causal constraint wherein **layout queries are prohibited from attending to image keys**, while image tokens retain full access to box tokens. Formally:

$$A_{ij} = \begin{cases} 0 & \text{if } i \in \mathcal{S}_{\mathrm{box}} \text{ and } j \in \mathcal{S}_{\mathrm{img}}, \\ 1 & \text{otherwise.} \end{cases} \tag{8}$$

This blocks direct image-to-layout attention. The staged inference procedure establishes the planning order; the mask alone does not exclude indirect paths through other token streams.

### 3.3 Training Strategy

We couple and decouple the noise schedules of the three visual modalities to train one network for spatial planning, layout-guided rendering, and joint denoising. The resulting model is also used for editing, restoration, and exploratory authenticity analysis.

To optimize the multi-modal flow transformer, we define the per-modality loss as the mean squared error between the predicted velocity $\mathbf{v}_\theta$ and the target direction $\mathbf{u}_m = \boldsymbol{\epsilon}_m - \mathbf{m}_0$:

$$\mathcal{L}_m(t_m) = \mathbb{E}_{t_m, \mathbf{m}_0, \boldsymbol{\epsilon}_m}\left[\|\mathbf{v}_\theta(\mathbf{Z}_t, t_m, \dots) - (\boldsymbol{\epsilon}_m - \mathbf{m}_0)\|^2\right], \tag{9}$$

where $m \in \{\mathrm{img}, \mathrm{box}, \mathrm{cond}\}$ represents the respective modality. At each iteration we sample one of four regimes from a probability distribution $\mathcal{P} = \{p_{\mathrm{S1}}, p_{\mathrm{S2}}, p_{\mathrm{joint}}, p_{\mathrm{uncond}}\}$. A regime is fully specified by the noise state it assigns to the three visual streams, summarized in Tab. 6. Here $\sigma(\cdot)$ is a sigmoid warp of the normal distribution that concentrates sampling on intermediate timesteps for better flow estimation; $t_m = 0$ means the stream is given clean, and $t_m = 1$ means it is pure noise and carries no clean-sample information.

The regimes are four views of one network rather than four trainings. S1 sees a pure-noise image and plans layout from text and the standard-font condition; S2 sees a clean layout and learns to render under it, as it does when a user supplies edited boxes. The Joint regime corrupts all three visual streams at a shared level $t$, supporting restoration from partial observations. The last regime removes the visual layout and font conditions, providing the image branch used for classifier-free guidance; text conditioning remains available.

**Box Perturbation.** In S2 we add a perturbation $\delta$ to the clean box latent. This exposes the model to variation in layout conditioning. During generation, S2 receives predicted layouts; during editing, it receives user-specified layouts. Perturbation is intended to improve tolerance to layout errors. The comparison in Sec. 4.4 does not isolate its contribution from that of shared model parameters.

Table 1: **Quantitative comparison of calligraphy generation methods.** CLIP/GPT scores are reference-based proxies for stylistic similarity. CharsAcc is the F1 score from PP-OCR (Cui et al., 2025).

| Method | CLIP-Score↑ | GPT-Score↑ | CharsAcc↑ |
|---|---|---|---|
| FontDiffuser | 0.8029 | 2.4 | **75.52%** |
| Gemini-Nano2 | 0.7620 | 2.2 | 69.23% |
| Hunyuan 3.0 | 0.7547 | 2.3 | 38.47% |
| Seedream 4.0 | 0.7864 | 2.6 | 62.54% |
| GPT-Image-1 | 0.7897 | 2.7 | 50.18% |
| UniCalli | 0.9051 | 3.6 | 42.03% |
| **CalliMaster** (pred. box) | 0.9574 | 3.9 | 33.41% |
| **CalliMaster** (given box) | **0.9663** | **4.1** | 33.56% |
| GT (Reference) | – | – | 34.12% |

Table 2: **User study results** (5-point Likert, mean, std; 15 evaluators). All five criteria: Tab. 7.

| **Method** | **Style Fidelity↑** | **Layout Similarity↑** | **Overall↑** |
|---|---|---|---|
| FontDiffuser | 1.720, 1.385 | 1.493, 1.287 | 1.840, 1.346 |
| GPT-Image-1 | 3.147, 1.178 | 2.360, 1.194 | 2.680, 1.253 |
| Hunyuan 3.0 | 2.653, 1.268 | 2.573, 1.136 | 3.013, 1.108 |
| Gemini-Nano2 | 2.373, 1.312 | 2.187, 1.224 | 2.453, 1.285 |
| UniCalli | 4.187, 0.968 | 4.080, 0.914 | 4.467, 0.812 |
| **CalliMaster (Ours)** | **4.427, 0.856** | **4.520, 0.738** | **4.680, 0.674** |

### 3.4 Inference and Downstream Applications

Generation, editing, and restoration use one ODE integrator $\Psi(\mathbf{z}_1, t_s \to t_e; \mathcal{C})$, which evolves a latent $\mathbf{z}$ from $t_s$ to $t_e$ conditioned on the context $\mathcal{C}$.

**Controllable Generation.** Standard synthesis follows the cascaded protocol of Sec. 3.3: the layout is planned first, $\hat{\mathbf{b}}_0 = \Psi(\mathbf{b}_1, 1 \to 0; \{\mathbf{c}, \text{txt}\})$, then the image is synthesized under it, $\hat{\mathbf{x}}_0 = \Psi(\mathbf{x}_1, 1 \to 0; \{\mathbf{c}, \hat{\mathbf{b}}_0, \text{txt}\})$.

**Layout-guided Editing.** Users modify $\mathbf{b}$ to $\mathbf{b}'$, updating text and the font condition when character content changes. Stage 2 then renders the edited conditions directly:

$$\hat{\mathbf{x}}'_{\text{edited}} = \Psi(\mathbf{x}_1, 1 \to 0; \{\mathbf{c}', \mathbf{b}', \text{txt}'\}). \tag{10}$$

Reusing the initial noise $\mathbf{x}_1$ keeps the random input fixed while the edited conditions guide brushwork and inter-character connections. Original stroke texture and ink variation need not remain unchanged.

**Artifact Restoration.** Because the Joint regime denoises layout and image at a shared noise level, the same model can be used for damaged pages. Given a corrupted image and a binary mask, joint inference repeatedly re-imposes observed content outside the mask while estimating the missing content. The masked-inference formulation and results under simulated damage are given in Sec. F.1.

**Exploratory Authenticity Analysis.** The learned velocity field provides a reconstruction-based descriptor of a page relative to the trained model. We add noise at several levels and aggregate single-step reconstruction errors into a *Diffusion Reconstruction Score* (DRS). Errors at different noise levels may respond to different visual scales. The protocol and score definition are in Sec. F.2; results appear in Sec. 4.5.

Table 3: **Ablation studies.** Both panels use the matched 72-hour reduced-budget checkpoint with predicted layouts, with lower CLIP than the full-run model in Tab. 1. (a) Unified vs. separate models (Sec. D). (b) Component ablations.

| | *Image quality* | | | *Layout* | | |
|---|---|---|---|---|---|---|
| **Method** | CLIP↑ | LPIPS↓ | GPT↑ | Layout EMD↓ | Rhythm EMD↓ | Layout–Ink Align↑ |
| *(a) Unified vs. separate models* | | | | | | |
| **CalliMaster** (unified) | **0.923** | **0.177** | **4.1** | **0.013** | **0.030** | $\mathbf{0.829 \pm 0.13}$ |
| Separate S1 + S2 | 0.871 | 0.258 | 3.4 | 0.019 | 0.042 | $0.761 \pm 0.16$ |
| *(b) Component ablations* | | | | | | |
| **CalliMaster** | **0.9234** | **0.1768** | – | – | – | – |
| w/o Box Prediction | 0.8698 | 0.2149 | – | – | – | – |
| w/o Modulate Embedding | 0.7400 | 0.3782 | – | – | – | – |

# 4 Experiments

## 4.1 Implementation Details

We use Flux (Labs et al., 2025) with frozen T5-XXL (Raffel et al., 2023) and VAE encoders, trained on 8×H100 GPUs with batch size 8. Half the training samples are synthetic ($p_{\text{synth}}$=0.5; Sec. D). Inference uses 25 rectified-flow steps per stage. Architecture, optimization, regime probabilities, and preprocessing details are in Secs. A and B.

**Evaluation Metrics.** CLIP-Score and GPT-Score compare images against references, LPIPS measures perceptual distance, and CharsAcc estimates OCR readability. We additionally evaluate composition using *Layout EMD*, a Wasserstein distance over box geometry; *Rhythm EMD*, the corresponding distance over adjacent spacing, scale variation, and page density; and *Layout–Ink Align*, the fraction of ink inside planned boxes. Feature definitions and protocols are in Sec. C.2.

## 4.2 Comparison with State-of-the-Art Methods

We compare against specialized calligraphy models (UniCalli (Xu et al., 2026), FontDiffuser (Yang et al., 2024)) and general-purpose foundational models (GPT-Image-1 (OpenAI, 2025), Hunyuan 3.0 (Cao et al., 2025), Seedream 4.0 (Team Seedream, 2025), Gemini-Nano2 (Google DeepMind, 2026)). All methods target the same text and style, but their available conditioning inputs differ. CalliMaster uses *predicted* rather than ground-truth layouts throughout Fig. 3. Per-method prompts, versions and inference settings: Sec. C.4.

**Quantitative Evaluation.** With predicted boxes, CalliMaster scores $0.9574$ on CLIP-Score and $3.9$ on GPT-Score (Tab. 1). Both scores exceed those of the listed baselines. Given boxes increase them to $0.9663$ and $4.1$. CharsAcc changes from $33.41\%$ with predicted boxes to $33.56\%$ with given boxes. FontDiffuser has the highest CharsAcc ($75.52\%$). The authentic reference scores $34.12\%$, indicating that the OCR metric also struggles with real cursive calligraphy. CalliMaster's proximity to the reference score alone does not establish glyph correctness or artistic quality. We therefore also report the perceptual evaluation below.

**Qualitative Evaluation.** In the examples of Fig. 3, several baselines show character omissions or structural errors; CalliMaster better preserves the intended page composition in these cases.

**User Study.** Fifteen evaluators rated randomized, anonymized outputs on five criteria (5-point Likert; background and protocol in Sec. C.3). CalliMaster scores highest overall ($4.68$) and leads in style fidelity, layout similarity (Tab. 2) and naturalness. Its mean glyph-accuracy rating is $4.85$, while FontDiffuser's is $4.92$; these descriptive results do not establish equivalence.

## 4.3 Layout-guided Editing

The intermediate boxes serve as geometric prompts for replacement, scaling, repositioning, deletion, and insertion (Fig. 4). Stage 2 regenerates the page under the edited layout, adapting visible stroke connections while retaining the supplied positions of unedited boxes. Layout-to-box conversion and further examples appear in Secs. B.4 and E.1.

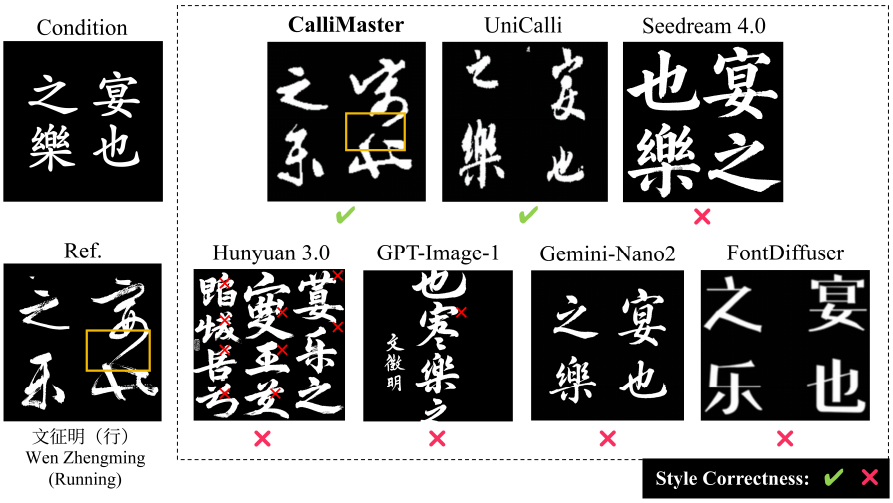



Figure 3: **Qualitative comparison of CalliMaster against state-of-the-art models.** Small red crosses mark character errors; large crosses and green checks indicate the authors' style judgments. Yellow boxes □ highlight continuous strokes in CalliMaster and the reference.

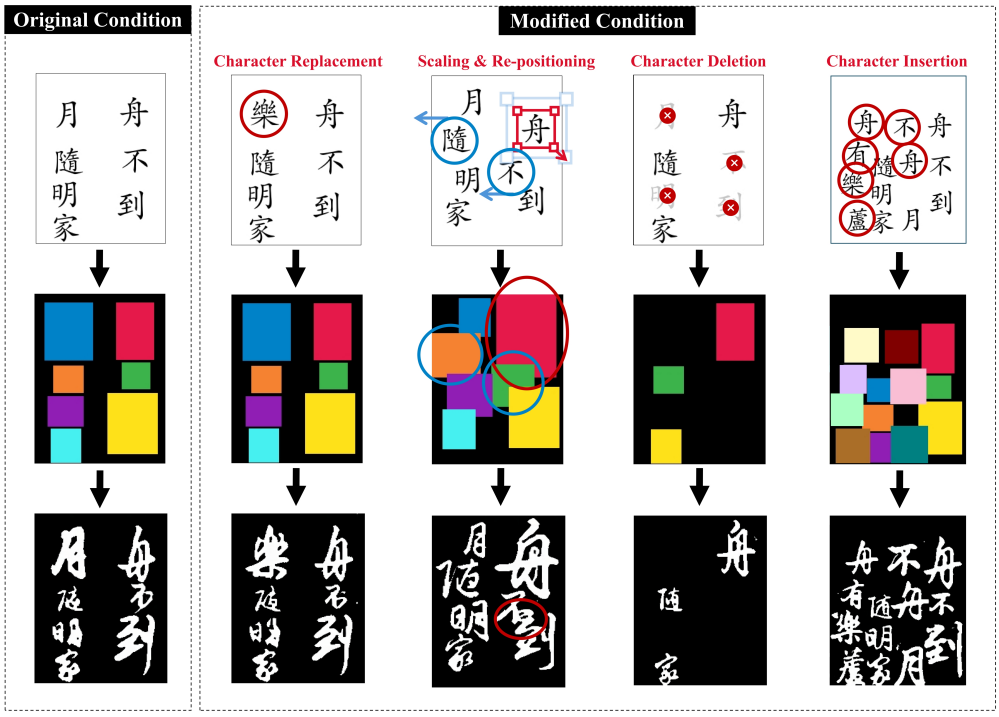



Figure 4: **Layout-guided editing with geometric prompts.** Edited boxes specify character regions for Stage 2 regeneration; the resulting page adapts visible stroke connections to the supplied layout.

### 4.4 Ablation Study

**Unified vs. Separate Models.** We compare CalliMaster with two independently trained stage models under a matched 72-hour training budget on the same hardware, shared across stages for the separate pipeline. Both use predicted layouts at inference. In Tab. 3(a), the unified model achieves higher CLIP (0.923 vs. 0.871) and lower LPIPS (0.177 vs. 0.258); the separate pipeline has 46% higher Layout EMD, 40% higher Rhythm EMD, and lower Layout–Ink Align. This comparison evaluates the complete training recipe: parameter sharing, regime mixing, and layout perturbation

Table 4: **DRS across authentic, perturbed, and generated pages.** L/M add 8/20 blob artifacts to authentic pages. Lower is better under this reconstruction score.

| Page | DRS↓ | Page | DRS↓ |
|---|---|---|---|
| GT | **0.186** | FontDiffuser | 0.561 |
| GT+Noise(L) | 0.237 | Gemini-Nano2 | 0.826 |
| GT+Noise(M) | 0.472 | GPT-Image-1 | 0.962 |
| CalliMaster | 0.299 | Seedream 4.0 | 0.997 |

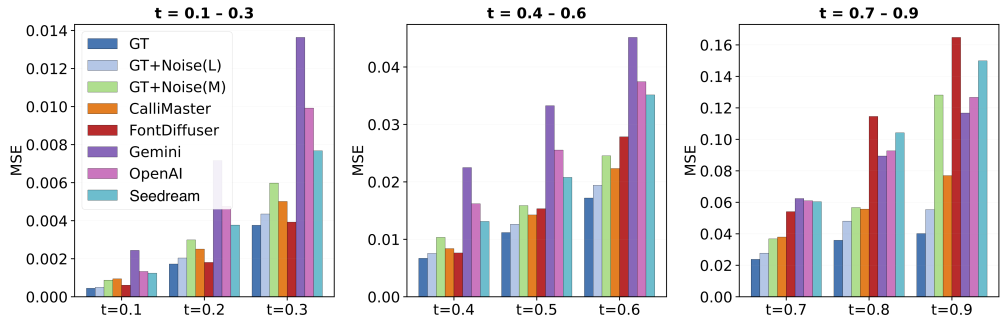



Figure 5: **DRS by noise scale.** Reconstruction errors across nine noise levels; their relationship to particular visual properties remains exploratory. The legend abbreviates Gemini-Nano2, GPT-Image-1, and Seedream 4.0.

differ, so their individual effects are not isolated. Timing and checkpoint storage are reported in Sec. D.

**Key Components.** Panel (b) of Tab. 3 reports component ablations against the same reduced-budget CalliMaster reference. Removing box prediction lowers CLIP from $0.9234$ to $0.8698$ and increases LPIPS from $0.1768$ to $0.2149$. Removing the modulate embedding yields CLIP $0.7400$ and LPIPS $0.3782$. These results support the contribution of both components under the evaluated settings. The synthetic-data proportion $p_{\text{synth}}$ is analyzed in Sec. D.

### 4.5 Exploratory Authenticity Analysis

DRS asks how well the learned flow reconstructs a query page after controlled corruption. We encode the page, add noise at level $t$, and estimate its clean latent with one flow step, $\hat{\mathbf{x}}_0 = \mathbf{x}_t - t\mathbf{v}_\theta$. We compare this estimate with the query latent using MSE and aggregate errors over nine levels ($t = 0.1, \ldots, 0.9$), with three trials at each level. A lower score means that the model reconstructs the page more accurately under the supplied conditions. The full formulation is in Sec. F.2.

The two comparisons test different aspects of this signal. Adding 8 or 20 blobs to authentic pages raises DRS from $0.186$ to $0.237$ and $0.472$ (Tab. 4), showing sensitivity to the tested damage. Across generators, CalliMaster scores lowest ($0.299$). Figure 5 reveals differences hidden by this aggregate: FontDiffuser is competitive at low noise but separates from CalliMaster at higher noise. Comparisons should be made within each noise level, since the reconstruction error also depends on $t$ and the panels use different vertical scales. Such profiles may help characterize deviations from learned calligraphic patterns, but do not identify specific stroke or layout defects. Because the evaluator is CalliMaster itself, its outputs may be favored; authentic–counterfeit discrimination still requires independent, expert-labeled validation.

## 5 Conclusion

CalliMaster unifies page-layout planning and rendering in one transformer, with rasterized boxes serving as editable visual prompts. Under a matched training budget, it improves image and layout metrics over separately trained stages. Restoration under simulated damage and exploratory DRS analysis extend its applications. Broader artistic text domains and independent authenticity validation remain future work.

## AI USE STATEMENT

In this paper, we used LLMs, such as ChatGPT and Claude, for language polishing and writing refinements (grammar, academic writing, and translation), for assistance in implementing parts of the method code, and for building the supplementary web page. We also used generative AI models as experimental baselines and a vision–language model to compute GPT-Score, as described in the experimental sections and appendix. We did not use AI tools to generate synthetic data or to clean or annotate datasets. The authors were solely responsible for the core ideas, research problem, method design, analyses, and experimental results; all AI-assisted code was reviewed and tested by the authors. We reviewed all AI-assisted work and take responsibility for the final content of this paper.

## ETHICS STATEMENT

The dataset was created from digitized historical works that are mainly publicly available. We also generated additional synthetic data from publicly available materials. In this paper, human evaluation was conducted for assessing the quality of generated calligraphy images, which did not involve sensitive personal data. The generated images may lead to copyright or cultural attribution concerns. Therefore, our research is intended for academic purposes, with the aim of preserving and supporting cultural heritage.

## REPRODUCIBILITY STATEMENT

Full model and training details are given in Sec. 4.1 and Sec. B, covering the backbone configuration, optimizer settings, the per-regime objectives and sampling probabilities $\mathcal{P}$, the auxiliary loss weight $\lambda_{\text{aux}}$, and the box perturbation range. Data sources, annotation and the synthetic generation pipeline are described in Sec. A. All metric definitions, the user-study protocol and the per-baseline prompts and inference settings are in Sec. C and Sec. C.4. We will release code, model weights and our bounding-box annotations upon acceptance.

## REFERENCES

Siyu Cao, Hangting Chen, Peng Chen, Yiji Cheng, Yutao Cui, Xinchi Deng, Ying Dong, Kipper Gong, Tianpeng Gu, Xiusen Gu, et al. Hunyuanimage 3.0 technical report. *arXiv preprint arXiv:2509.23951*, 2025.

Boyuan Chen, Diego Marti Monso, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser-2: Unleashing the power of language models for text rendering, 2023. URL https://arxiv.org/abs/2311.16465.

Zhenlian Chen. *Calligraphy Aesthetics*. Shanghai Book and Painting Publishing House, 2017. ISBN 9787547915936.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, Yue Zhang, Wenyu Lv, Kui Huang, Yichao Zhang, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, and Yanjun Ma. Paddleocr 3.0 technical report, 2025. URL https://arxiv.org/abs/2507.05595.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Google DeepMind. Nano banana 2: Gemini 3.1 flash image. https://deepmind.google/models/gemini-image/, 2026.

Haibin He, Xinyuan Chen, Chaoyue Wang, Juhua Liu, Bo Du, Dacheng Tao, and Yu Qiao. Diff-font: Diffusion model for robust one-shot font generation. *International Journal of Computer Vision*, 2024. doi: 10.1007/s11263-024-02137-0. URL https://arxiv.org/abs/2212.05895.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Qiyao Hu, Yinyin Luo, Xianlin Peng, Rui Cao, Jinye Peng, and Jianping Fan. Calliecd: An error correction diffusion model for multi-style chinese calligraphy generation. *Pattern Recognition*, 180(Part D):114373, 2026. ISSN 0031-3203. doi: 10.1016/j.patcog.2026.114373. URL https://doi.org/10.1016/j.patcog.2026.114373.

Black Forest Labs, Stephen Batifol, Andreas Blattmann, Frederic Boesel, Saksham Consul, Cyril Diagne, Tim Dockhorn, Jack English, Zion English, Patrick Esser, Sumith Kulal, Kyle Lacey, Yam Levi, Cheng Li, Dominik Lorenz, Jonas Müller, Dustin Podell, Robin Rombach, Harry Saini, Axel Sauer, and Luke Smith. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Haolong Li, Chenghao Du, Ziheng Jiang, Yifan Zhang, Jiawei Ma, and Chen Ye. Towards automated chinese ancient character restoration: A diffusion-based method with a new dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 3073–3081, 2024.

Qisheng Liao, Zhinuo Wang, Muhammad Abdul-Mageed, and Gus Xia. Callipaint: Chinese calligraphy inpainting with diffusion model. 2023. URL https://arxiv.org/abs/2312.01536.

Qisheng Liao, Liang Li, Yulang Fei, and Gus Xia. Calliffusionv2: Personalized natural calligraphy generation with flexible multi-modal control. 2024. URL https://arxiv.org/abs/2410.03787.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11461–11471, 2022.

OpenAI. Addendum to gpt-4o system card: Native image generation. https://cdn.openai.com/11998be9-5319-4302-bfbf-1167e093f1fb/Native_Image_Generation_System_Card.pdf, 2025.

Wei Pan, Huiguo He, Hiuyi Cheng, Yilin Shi, and Lianwen Jin. Diffink: Glyph- and style-aware latent diffusion transformer for text to online handwriting generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=XKOEQFKFdL.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Zhenzhong Qiu. *Form and interpretation of calligraphy*. SDX Joint Publishing Company, 2022. ISBN 9787108073167.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Team Seedream. Seedream 4.0: Toward next-generation multimodal image generation. *arXiv preprint arXiv:2509.20427*, 2025.

Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=ezBH9WE9s2.

Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018.

Shan-Jean Wu, Chih-Yuan Yang, and Jane Yung-jen Hsu. Calligan: Style and structure-aware chinese calligraphy character generator. *arXiv preprint arXiv:2005.12500*, 2020.

Yangchen Xie, Xinyuan Chen, Li Sun, and Yue Lu. Dg-font: Deformable generative networks for unsupervised font generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/papers/Xie_DG-Font_Deformable_Generative_Networks_for_Unsupervised_Font_Generation_CVPR_2021_paper.pdf.

Tianshuo Xu, Kai Wang, Zhifei Chen, Leyi Wu, Tianshui Wen, Fei Chao, and Ying-Cong Chen. Unicalli: A unified diffusion framework for column-level generation and recognition of chinese calligraphy. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=OSIPdrw56X.

Zhenhua Yang, Dezhi Peng, Yuxin Kong, Yuyi Zhang, Cong Yao, and Lianwen Jin. Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and style contrastive learning. In *Proceedings of the AAAI conference on artificial intelligence*, 2024.

Liguo Zhang, Yalong Zhu, Achref Benarab, Yusen Ma, Yuxin Dong, and Jianguo Sun. Dp-font: Chinese calligraphy font generation using diffusion model and physical information neural network. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI 2024)*, pp. 7796–7804, 2024. doi: 10.24963/ijcai.2024/863. URL https://www.ijcai.org/proceedings/2024/863.

# APPENDIX CONTENTS


A **Data Preparation** 14

B **Model and Training Details** 14

B.1 Configuration 14

B.2 Per-Regime Training Objectives 15

B.3 Layout Representation 15

B.4 Interactive Layout Editing 16

C **Evaluation Protocol** 16

C.1 Image-Level and Perceptual Metrics 16

C.2 Layout-Level Metrics 16

C.3 User Study Protocol 17

C.4 Baseline Protocols and Prompts 18

D **Additional Ablations and Efficiency** 20

D.1 Unified vs. Separate Models: Setup 20

D.2 Training and Inference Efficiency 20

D.3 Synthetic Data Proportion 20

E **The Layout as a Semantic Plan: Extended Results** 21

E.1 Layout-guided Editing 21

E.2 Style-Conditioned Spatial Layout and Generation 22

E.3 Additional Generated Samples 24

F **Training-Free Applications of the Joint Regime** 27

F.1 Artifact Restoration 28

F.2 Authenticity Analysis via the Diffusion Reconstruction Score 28

The supplementary material includes an anonymous, self-contained web page (`project_page/index.html`; it opens locally in any browser and needs no internet connection). It replays the pipeline of Fig. 1 step by step, shows pages generated in several calligraphers' styles, shows the box-editing examples of Sec. E.1, and lets the reader browse the reported results by experiment and metric. All images and numbers on the page come from this paper; the page runs no inference.

# A Data Preparation

**Realistic Data.** We integrate the public UniCalli dataset (Xu et al., 2026) with a newly collected dataset of authentic calligraphy works sourced from the internet, spanning diverse scripts (Kai/Regular, Xing/Running, Cao/Cursive) and historical periods. For the newly collected images, we perform manual annotation to provide per-character bounding boxes and Unicode labels. Given that raw web data frequently suffers from background clutter and paper degradation, we implement a specialized U-Net based denoising model to enhance stroke clarity. Following the training paradigm of ESRGAN (Wang et al., 2018), we optimize the model using adversarial and perceptual losses. The training data is synthesized by injecting random degradation artifacts into clean samples. Characters are clustered into columns, filtered for spatial continuity, and cropped into page-level samples. The condition image $\mathbf{c}$ is rendered using standard fonts, while the layout $\mathbf{b}$ encodes character regions with random colors. During training, both are derived from the annotated boxes, so each supervised triplet has spatially aligned $\mathbf{c}$ and $\mathbf{b}$; the synthetic pipeline below shares this property. At inference, Stage 1 conditions on the standard-font image and the style description to predict character scale and placement (Sec. E.2). The condition image therefore supplies initial spatial information as well as glyph structure.

**Synthetic Data.** Following UniCalli (Xu et al., 2026), we construct a synthetic data pipeline. For each sample, we randomly select $k \in [3, 20]$ characters and generate layouts using four modes: *grid*, *column*, *random*, and *scatter*. We apply per-character scale jitter $s \sim \mathcal{U}(0.85, 1.15)$ and aspect ratio perturbation $r \sim \mathcal{U}(0.9, 1.1)$. High-quality calligraphy TTF fonts are rendered into these boxes, yielding paired triplets (image, condition, box) with pixel-perfect alignment.

**Merging Strategy and Robustness.** Each batch element is drawn from either the realistic or synthetic pool via Bernoulli sampling with probability $p_{\text{synth}}$. Synthetic data provides aligned box–image pairs and exposes the renderer to perturbed layouts resembling user edits. This variation is intended to improve tolerance to irregular box configurations, although the experiments do not establish generalization to every possible edit.

# B Model and Training Details

This section completes Secs. 3.3 and 4.1 of the main paper. It lists the full configuration, writes out the four per-regime objectives, and then describes the layout modality end to end: how character boxes become transformer tokens, and how a predicted layout becomes a set of boxes the user can edit.

## B.1 Configuration

Tab. 5 collects the architecture, optimization and sampling settings. The backbone, the text encoder and the VAE are all inherited from the pre-trained Flux model (Labs et al., 2025); only the transformer blocks, the shared visual input projection $W_{\text{in}}$ and the modality embeddings $\mathbf{e}_m$ are trained. Notably, no modality-specific encoder is introduced for the layout stream, which is what allows layout to be added to a pre-trained image DiT while reusing its visual encoder.

Table 5: **Full training and inference configuration.**

| Setting | Value |
|---|---|
| Backbone | Flux (Labs et al., 2025), $d_{\text{model}} = 3072$ |
| Transformer blocks | 19 double-stream + 38 single-stream |
| Text encoder | T5-XXL (Raffel et al., 2023), frozen |
| Visual encoder / decoder | pre-trained VAE, frozen, shared by img / box / cond |
| Trained parameters | transformer blocks, $W_{\text{in}}$, modality embeddings $\mathbf{e}_m$ |
| Position encoding | 3D RoPE (Su et al., 2024) over (modality index, $y$, $x$) |
| Hardware | 8×H100 |
| Optimizer | AdamW, lr $= 1\times10^{-5}$ |
| Batch size | 8 |
| Regime probabilities $\mathcal{P}$ | $p_{\text{S1}} = 0.35$, $p_{\text{S2}} = 0.35$, $p_{\text{joint}} = 0.25$, $p_{\text{uncond}} = 0.05$ |
| Auxiliary loss weight | $\lambda_{\text{aux}} = 0.01$ |
| Box perturbation | $\delta \sim \mathcal{U}(0, 0.1)$ on the clean box latent (S2 only) |
| Timestep sampling | $t \sim \sigma(\mathcal{N}(0, 1))$, $\sigma$ the logistic sigmoid |
| Synthetic data proportion | $p_{\text{synth}} = 0.5$ |
| Characters per sample | up to 20 |
| Canonical font size | 64 px |
| Inference solver | rectified flow ODE, 25 steps *per stage* (50 total) |

### B.2 Per-Regime Training Objectives

Table 6: **The four training regimes.** Each iteration samples one, which fixes the noise state of the three visual streams and thereby what the shared network learns. Non-primary streams are kept as auxiliary terms weighted by $\lambda_{\text{aux}}$, except in the Joint regime, which weights all three equally. Complete objectives: Sec. B.

| Regime | $t_{\text{img}}$ | $t_{\text{box}}$ | $t_{\text{cond}}$ | Primary objective | Inference mode it enables |
|---|---|---|---|---|---|
| S1: spatial planning | 1 | $\sigma(\mathcal{N}(0,1))$ | 0 | $\mathcal{L}_{\text{box}}$ | layout prediction from text and font image |
| S2: content synthesis | $\sigma(\mathcal{N}(0,1))$ | $0\ (+\delta)$ | 0 | $\mathcal{L}_{\text{img}}$ | layout-guided rendering, editing |
| Joint | $t$ | $t$ | $t$ | $\mathcal{L}_{\text{img}}+\mathcal{L}_{\text{box}}+\mathcal{L}_{\text{cond}}$ | holistic restoration / inpainting |
| Unconditional | $\sigma(\mathcal{N}(0,1))$ | 1 | 1 | $\mathcal{L}_{\text{img}}$ | classifier-free guidance |

With the per-modality loss $\mathcal{L}_m$ of Eq. (9), the four regimes of Tab. 6 correspond to the following objectives. In S1 the layout term leads and the other two streams act as auxiliary regularizers:

$$\mathcal{L}_{\text{S1}} = \mathcal{L}_{\text{box}} + \lambda_{\text{aux}}\,(\mathcal{L}_{\text{img}} + \mathcal{L}_{\text{cond}}), \tag{11}$$

while S2 mirrors this arrangement with the image term in the lead, the layout supplied clean but perturbed by $\delta$:

$$\mathcal{L}_{\text{S2}} = \mathcal{L}_{\text{img}} + \lambda_{\text{aux}}\,(\mathcal{L}_{\text{box}} + \mathcal{L}_{\text{cond}}). \tag{12}$$

The joint regime corrupts all three visual streams at a single shared level $t_{\text{img}} = t_{\text{box}} = t_{\text{cond}} = t$ and weights their reconstruction equally, which is what lets the model recover layout and content simultaneously from a partial observation:

$$\mathcal{L}_{\text{joint}} = \mathcal{L}_{\text{img}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{cond}}. \tag{13}$$

The unconditional regime sets $t_{\text{box}} = t_{\text{cond}} = 1$ so that neither layout nor condition carries information, and uses the image loss as its primary term, retaining auxiliary losses:

$$\mathcal{L}_{\text{uncond}} = \mathcal{L}_{\text{img}} + \lambda_{\text{aux}}\,(\mathcal{L}_{\text{box}} + \mathcal{L}_{\text{cond}}). \tag{14}$$

This last regime supplies a branch without the visual layout and font conditions for classifier-free guidance. Text conditioning remains available, so "unconditional" refers here to the visual streams rather than to all inputs.

### B.3 Layout Representation

A layout is a set of $k$ axis-aligned character boxes $\{(x_1, y_1, x_2, y_2)_j\}_{j=1}^{k}$ in page pixel coordinates, obtained from manual annotation for realistic data and procedurally for synthetic data (Sec. A). Rather than feeding these coordinates to the network as a sequence, we render them: the boxes are painted onto a blank canvas at the page resolution, each character region $j$ filled with a color $\mathbf{r}_j$

drawn at random, producing a layout *image* $\mathbf{b}_{\text{img}}$. Random rather than fixed colors keep the model from tying a particular hue to a particular character, and keep neighboring boxes visually separable.

The layout image is then treated like the other visual streams. It passes through the *same* frozen VAE encoder used for the calligraphy image and the condition image, giving the clean layout latent $\mathbf{b}_0$; this is patchified with the shared projection $W_{\text{in}}$ into $N$ tokens, the modality embedding $\mathbf{e}_{\text{box}}$ is added, and 3D RoPE supplies the modality index together with the 2D position, yielding $\mathbf{z}_{\text{box}}^{1:N}$ in Eq. (5). There is no coordinate tokenizer, learned box head, or layout-specific encoder. The added modality embedding lets the fine-tuned transformer distinguish layout tokens while reusing the visual input pathway.

### B.4 Interactive Layout Editing

During spatial planning the layout tokens are initialized from noise and integrated to $\hat{\mathbf{b}}_0 = \Psi(\mathbf{b}_1, 1 \to 0; \{\mathbf{c}, \text{txt}\})$, with image tokens held at $t_{\text{img}} = 1$ and the causal mask of Eq. (8) blocking layout queries from reading image keys; decoding $\hat{\mathbf{b}}_0$ with the VAE decoder returns a predicted layout image. To make that prediction editable it is converted back into rectangles: pixels are grouped by color into connected regions, and each region contributes the axis-aligned rectangle that encloses it. Because every character region carries its own random color, the grouping separates characters even where their boxes touch.

The recovered rectangles are what the user manipulates. Moving, resizing, inserting, or deleting a box edits this coordinate list, which is held as external metadata. The edited list is re-rasterized and re-encoded, and the resulting latent is supplied *clean* ($t_{\text{box}} = 0$) to the content synthesis stage. Stage 2 regenerates the page under this fixed geometry; unedited boxes do not move unless the user moves them. S2 training with perturbed layout latents (Sec. 3.3) is intended to improve tolerance to irregular boxes. Edits that change the character sequence also require a matching update to the text and standard-font condition.

## C Evaluation Protocol

This section describes the evaluation procedures used in the main paper: the metrics, the user-study procedure, and the exact conditions under which each baseline was run.

### C.1 Image-Level and Perceptual Metrics

CLIP-Score and GPT-Score compare generated pages with the reference: the former uses embedding similarity, and the latter a vision–language model rating. They are proxies for style similarity because page content and composition can also affect the scores. LPIPS measures perceptual distance to the reference and is reported wherever a paired reference exists, as in the ablations of Sec. D. CharsAcc is the character-level F1 score of PP-OCR (Cui et al., 2025) on the generated page. Authentic cursive calligraphy is also difficult for OCR, but similarity to the reference's OCR score does not establish glyph correctness (Tab. 1). The layout metrics below evaluate spatial properties these image-level measures do not isolate.

### C.2 Layout-Level Metrics

Two of the three metrics compare *distributions* of geometric descriptors, because a calligraphic composition is not judged by per-character coordinate error against a reference; the third checks the internal consistency between a page's plan and its ink. None of the three scores inter-character connections, which resist automatic measurement; these are judged only in the user study (Sec. C.3).

**Layout EMD.** For each page we extract a set of geometric box features: character width and height, aspect ratio, box center position, inter-character gaps between neighbors within a column, the four page margins, and the negative-space ratio (the fraction of page area not covered by any box). Lengths and positions are expressed relative to the page dimensions so that pages of different sizes are comparable, and each feature is treated as a one-dimensional sample set. Layout EMD is

the Wasserstein-1 distance between the generated and reference sample sets, computed per feature and then aggregated across features into a single score; lower is better.

**Rhythm EMD.** Rhythm concerns how the composition *changes* across the page rather than its marginal statistics, so the features are ratios and densities instead of raw sizes: the ratio between consecutive inter-character gaps, the ratio between the scales of consecutive characters, and the occupancy of a $4\times4$ density grid laid over the page. Pages with similar marginal size and spacing distributions can differ in their sequential variation; Rhythm EMD is intended to capture this distinction. The distance and the aggregation are the same as above.

**Layout–Ink Align.** This metric asks whether rendered ink falls in the planned character regions. The generated page is binarized with Otsu's method to obtain an ink mask, and the score is the fraction of ink pixels inside the union of the character boxes. Unlike the two EMD metrics it needs no reference page, so it applies to edited as well as generated layouts. It is a geometric proxy: intended strokes crossing box boundaries lower the score, and oversized boxes can inflate it. We report mean $\pm$ std over the evaluated pages.

**Reporting.** All three metrics are computed over the full evaluation set, with no per-sample exclusion; the statistical unit for the reported mean and standard deviation is the page.

### C.3 User Study Protocol

We conducted a structured perceptual evaluation of 10 cases from two target calligraphers (Huang Tingjian and Sun Guoting). Each case presents outputs from 6 compared methods: FontDiffuser, GPT-Image-1, Hunyuan 3.0, Gemini-Nano2, UniCalli, and CalliMaster (Ours). The cases were shuffled in random order and presented without method labels, to reduce order and method-label biases in the ratings. We recruited 15 evaluators, including calligraphy enthusiasts and practicing professionals. Their experience with calligraphy ranges from 5 years to more than 30 years, and the group includes members of provincial calligraphy associations as well as senior practitioners. They rated each image on five dimensions (Style Fidelity, Glyph Accuracy, Naturalness, Layout Similarity, and Overall Quality) using a 5-point Likert scale. Every evaluator saw every case, so the reported means aggregate 15 ratings per case (N = 150 per method per criterion), and the reported standard deviations describe the spread of individual ratings rather than a test of inter-rater agreement. Tab. 7 gives all five criteria; the main paper's Tab. 2 reproduces three of them. Sample questionnaire interfaces are shown in Figs. 6 and 7.

Table 7: **Full user study results** across all five perceptual criteria (5-point Likert scale, mean, std). Best scores in **bold**.

| Method | Style Fidelity↑ | Glyph Accuracy↑ | Naturalness↑ | Layout Similarity↑ | Overall↑ |
|---|---|---|---|---|---|
| FontDiffuser | 1.720, 1.385 | **4.920, 0.396** | 2.080, 1.512 | 1.493, 1.287 | 1.840, 1.346 |
| GPT-Image-1 | 3.147, 1.178 | 3.720, 1.215 | 2.587, 1.302 | 2.360, 1.194 | 2.680, 1.253 |
| Hunyuan 3.0 | 2.653, 1.268 | 4.107, 0.984 | 3.187, 1.245 | 2.573, 1.136 | 3.013, 1.108 |
| Gemini-Nano2 | 2.373, 1.312 | 3.547, 1.278 | 2.840, 1.356 | 2.187, 1.224 | 2.453, 1.285 |
| UniCalli | 4.187, 0.968 | 4.760, 0.478 | 4.413, 0.782 | 4.080, 0.914 | 4.467, 0.812 |
| **CalliMaster (Ours)** | **4.427, 0.856** | 4.853, 0.384 | **4.613, 0.724** | **4.520, 0.738** | **4.680, 0.674** |

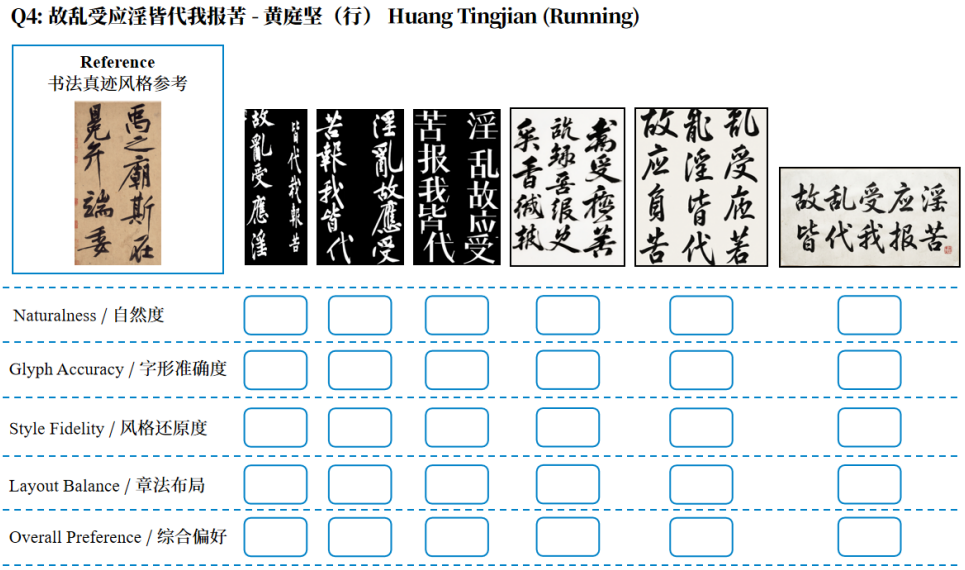



Figure 6: **Sample user study questionnaire for calligrapher Huang Tingjian.** This image displays one specific evaluation case. It presents 6 randomly ordered outputs from the compared methods. Evaluators rate each on Style Fidelity, Glyph Accuracy, Naturalness, Layout Similarity, and Overall Quality using a 5-point Likert scale.

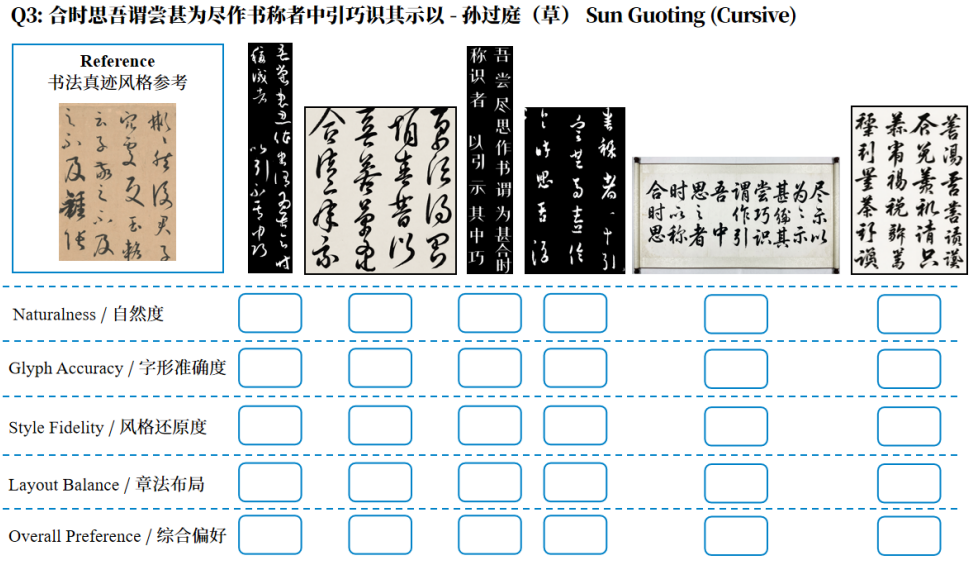



Figure 7: **Sample user study questionnaire for calligrapher Sun Guoting.** The same 10-case, 6-method, 5-criterion protocol as in Fig. 6 is applied, targeting a distinct calligraphic style.

### C.4 Baseline Protocols and Prompts

Tab. 8 records how each compared method was run. The conditioning is not identical across methods, because the methods do not accept identical inputs; we therefore state what each one actually received rather than claiming a uniform protocol. All methods are evaluated on the same 50 test samples and produce exactly one output per sample; no generation failed, and no output was resampled or selected. The general-purpose models were queried through their official APIs on 1–2 March 2026 with default settings, using only the text prompt below, as in their standard text-to-image usage; they did not receive the condition image **c**. The two specialized calligraphy baselines

were run with their official code, weights and default inference settings, under their own native conditioning protocols. Because conditioning differs across methods, the layout metrics are reported only for internal comparisons in which all systems receive identical conditioning; the cross-method comparison in Tab. 1 uses image-level metrics only.

Table 8: **Baseline protocols.** "Condition image" denotes the TTF-rendered reference $\mathbf{c}$ showing the target characters in a standard font. Every method produces one output for each of the same 50 test samples.

| Method | Access / version | Input actually given | Setting |
|---|---|---|---|
| **FontDiffuser** (Yang et al., 2024) | official code and weights | per character: target character + one authentic glyph of the target calligrapher as style reference; placed into CalliMaster's predicted layout | default |
| **UniCalli** (Xu et al., 2026) | official code and weights | target text in columns of up to five characters, script and calligrapher; columns concatenated into a page | default |
| **GPT-Image-1** (OpenAI, 2025) | OpenAI API, `gpt-image-1` | text prompt only | default |
| **Hunyuan 3.0** (Cao et al., 2025) | Tencent Cloud Hunyuan Image 3.0 API | text prompt only | default |
| **Seedream 4.0** (Team Seedream, 2025) | Volcano Engine Jimeng AI Image Generation 4.0 API | text prompt only | default |
| **Gemini-Nano2** (Google DeepMind, 2026) | Google Gemini API (Nano Banana 2) | text prompt only | default |
| **CalliMaster (ours)** | – | text + style instruction, condition image $\mathbf{c}$, predicted layout | Tab. 5 |

**Prompt for general-purpose models.** All four general-purpose models receive the same prompt, formed by appending the target text to the sample's caption from the dataset annotation. The caption specifies the background, the script, the desired structural qualities, and a short description of the target calligrapher:

```
{CAPTION} The image contains Chinese text writing:  "{TEXT}"
```

For example, test sample `032` is prompted with:

```
Traditional Chinese calligraphy works, background:  black,
font:  kai, symmetrical structure, stable center of gravity,
rigorous layout of the frame, clear gaps between strokes,
easy to identify the shape of the characters.  author:  Yan
Zhenqing (709-785), a Tang dynasty calligrapher and loyal
official, was a master of regular script, creating a robust,
upright style that emphasized moral integrity.  Works like
'Yan Qinli Stele' remain exemplary.  The image contains
Chinese text writing:  "学三篇内二篇记以"
```

**Specialized baselines.** FontDiffuser generates one character at a time, taking the target character as content input and a single authentic glyph by the target calligrapher as style reference. Its outputs are color-inverted to match the black page background and placed into the character boxes that CalliMaster predicts for the same sample, so FontDiffuser inherits CalliMaster's layout and its page-level scores reflect glyph rendering rather than layout planning. UniCalli generates one column of up to five characters at a time; the target text is split in reading order into consecutive five-character segments, the last segment holding the remaining characters, and the resulting columns are concatenated into a page.

# D Additional Ablations and Efficiency

## D.1 Unified vs. Separate Models: Setup

The comparison reported in the top block of Tab. 3 pits CalliMaster against a two-model pipeline in which spatial planning and content synthesis are handled by two independently trained networks. The separate S1 model is trained only under the S1 regime and the separate S2 model only under the S2 regime; both use the same backbone as CalliMaster, so the difference is the sharing of parameters and of the token space, not the capacity of a single stage.

The budget is matched at 72 h in total: the unified model spends all of it on one network, while the two-model pipeline splits it between its two stages. The layout-only S1 model is far cheaper per step (230 ms) than either model that must denoise dense image tokens ($\approx 610$ ms), so it completes more steps in its share (Tab. 9).

At inference the two pipelines run the same number of denoising steps per stage (25), so the unified model performs $2 \times 25$ steps in one loaded network while the two-model pipeline performs 25 steps in each of two networks loaded in turn. The separate S2 model is trained on ground-truth boxes but receives S1 predictions at test time. The unified S2 regime also receives predicted boxes at inference; its box perturbation during training may reduce sensitivity to this shift. The comparison does not isolate the effects of parameter sharing and box perturbation.

## D.2 Training and Inference Efficiency

Under the matched budget the unified model completes more optimization steps than either separate model, and at inference it is both faster end to end and half the size on disk. The compute subtotals show that the raw denoising work of the two-model pipeline ($5.8 + 15.2 = 21.0$ s) is actually below the unified model's 30.5 s, since the unified model carries layout, image and condition tokens through every forward pass; the end-to-end ordering reverses because of the doubled model I/O.

Table 9: **Training and inference efficiency under the matched 72 h budget.** Inference figures are per generated page. "Model I/O" is the cost of moving weights onto the accelerator, which the two-model pipeline pays twice because it must offload S1 before loading S2. The reported total latencies of 57 s and 65 s are end-to-end and therefore include this overhead.

| | **CalliMaster** | **Two-model S1** | **Two-model S2** |
|---|---|---|---|
| *Training* | | | |
| Per-step time | 610 ms | 230 ms | 608 ms |
| Steps completed | **400 k** | 350 k | 300 k |
| *Inference* (25 denoising steps per stage) | | | |
| Total denoising steps | 50 (2×25) | 25 | 25 |
| Compute subtotal | 30.5 s | 5.8 s | 15.2 s |
| Model I/O overhead | ∼27 s (load once) | ∼44 s (load S1 + offload + load S2) | |
| **Total latency** | **57 s** | **65 s** combined | |
| Checkpoint storage | 1× | 2× | |

## D.3 Synthetic Data Proportion

Fig. 8 shows the effect of the synthetic sampling probability $p_{\text{synth}}$ introduced in Sec. A. The reported CLIP and LPIPS results favor $p_{\text{synth}} = 0.5$, the setting used throughout the paper. Lower synthetic proportions may limit exposure to varied layouts, whereas higher proportions may favor font-like rendering over authentic ink variation. These are possible explanations of the trend, rather than effects isolated by the image metrics.

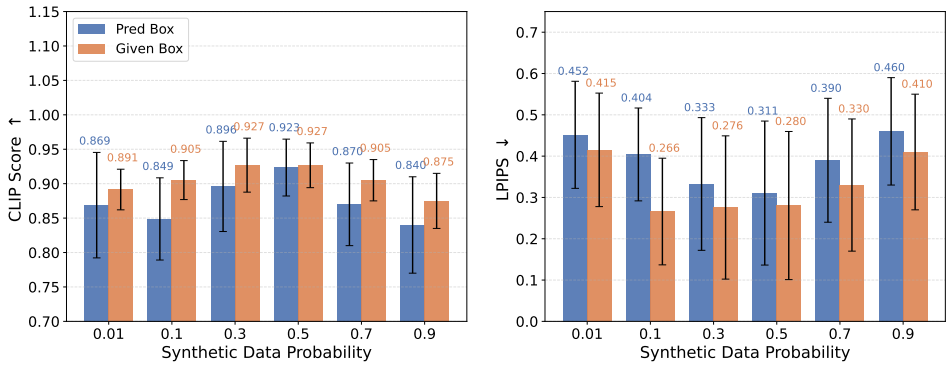



Figure 8: **Effect of Synthetic Data Proportion.** We analyze the impact of the synthetic data sampling probability ($p_{\text{synth}}$) on generation quality. The bar charts illustrate the CLIP Score (left) and LPIPS (right) across different training ratios, evaluated under both predicted and given bounding box settings.

# E THE LAYOUT AS A SEMANTIC PLAN: EXTENDED RESULTS

The results in this section illustrate how the layout guides rendering. Section E.1 changes boxes and shows the page regenerated under the edited geometry. Section E.2 shows predicted layouts for shared text under different style descriptions. Section E.3 presents further examples across calligraphers, scripts, and text content.

## E.1 LAYOUT-GUIDED EDITING

CalliMaster also treats intermediate bounding boxes as editing prompts. In Fig. 9, the supplied edited boxes guide page regeneration and the visible stroke connections. The box positions themselves are supplied by the user. The conversion between predicted layout images and editable rectangles is described in Sec. B.4.

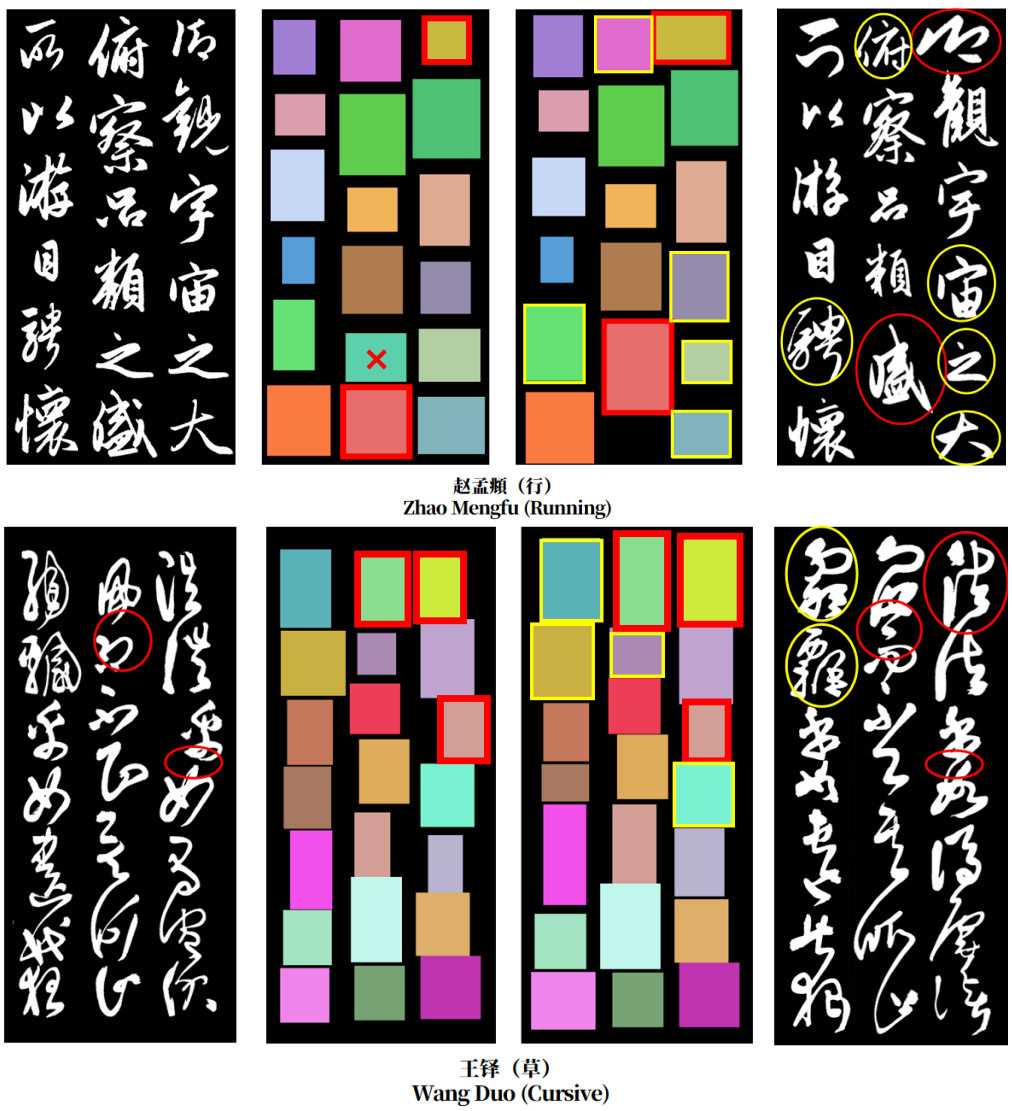



Figure 9: **Layout editing cases.** The two left images show an initial page and its box layout. Red outlines □ and crosses × mark user edits to the boxes. Stage 2 regenerates the page under the edited geometry. Red circles ◯ mark edited character regions; yellow markers highlight surrounding regions where the rendered ink changes.

### E.2 Style-Conditioned Spatial Layout and Generation

Figures 10–12 present three groups of results, each sharing the same text condition but targeting a different calligrapher and script style. For each group, the *top* image shows the spatial bounding-box layout inferred by CalliMaster for that particular calligraphic style, and the *bottom* image shows the corresponding generated calligraphy output. The examples show variation in rendered brushwork, character sizing, inter-character spacing, and column structure across the supplied style descriptions.

Because the text and random seed are fixed within each group, these differences come from the style description alone: style conditioning changes the predicted layout as well as the rendered strokes. The style description reaches the network through text tokens and the global embedding (Sec. 3.2).

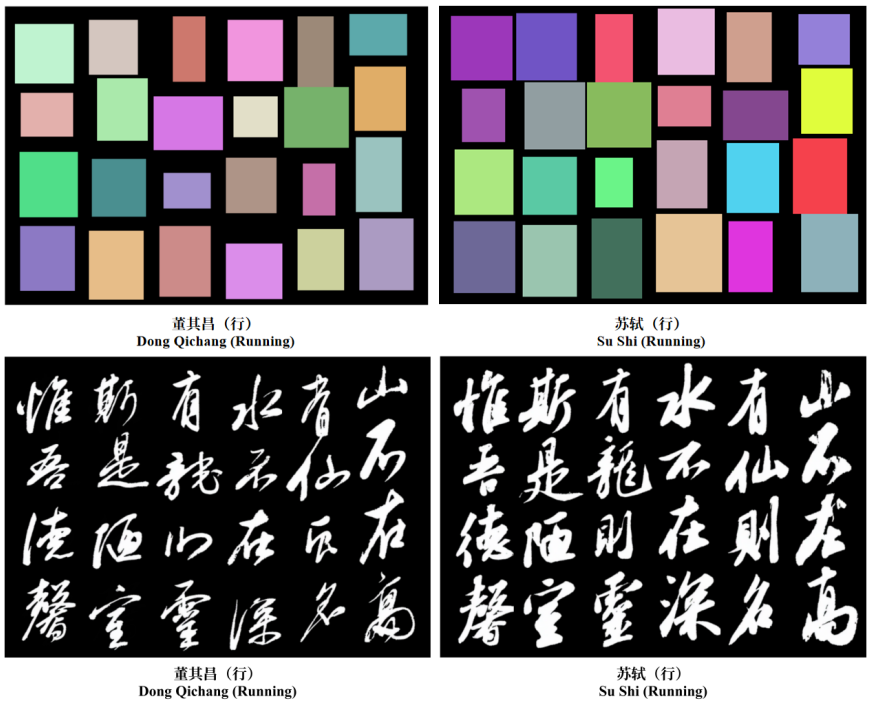



Figure 10: **Dong Qichang and Su Shi (running script). Top:** spatial bounding-box layouts generated for calligraphers' styles, showing the inferred character sizes, positions, and column arrangement. **Bottom:** calligraphy images generated from the layout above, exhibiting the brushwork and stroke character distinctive to these calligraphers.

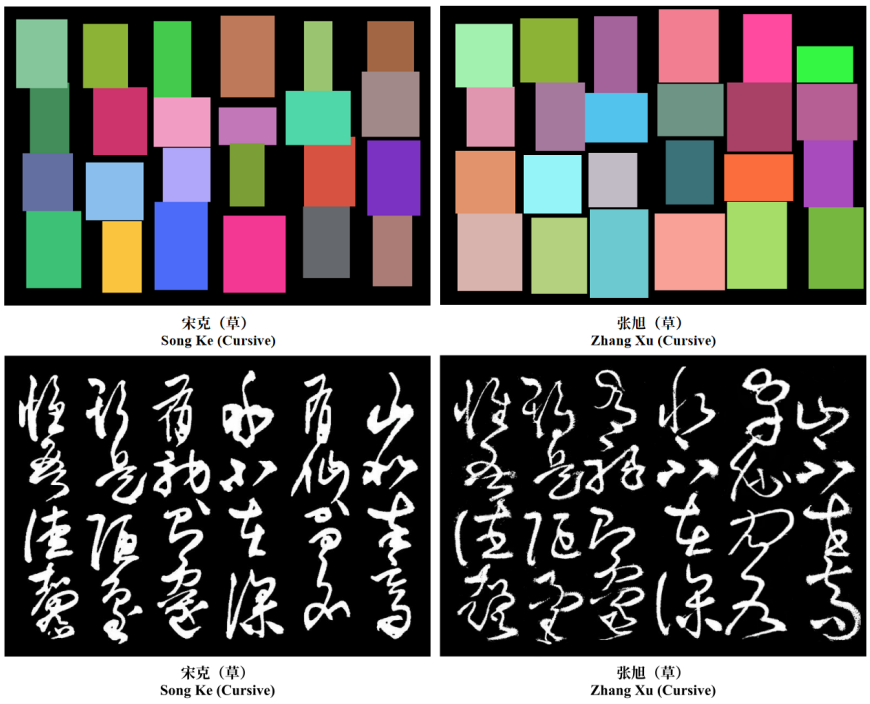



Figure 11: **Song Ke and Zhang Xu (cursive script). Top:** spatial layout adapted to the target styles under the exact same text condition. Note the distinct spatial rhythm and character proportions compared with group 1. **Bottom:** generated calligraphy reflecting these calligraphers' individual aesthetics.

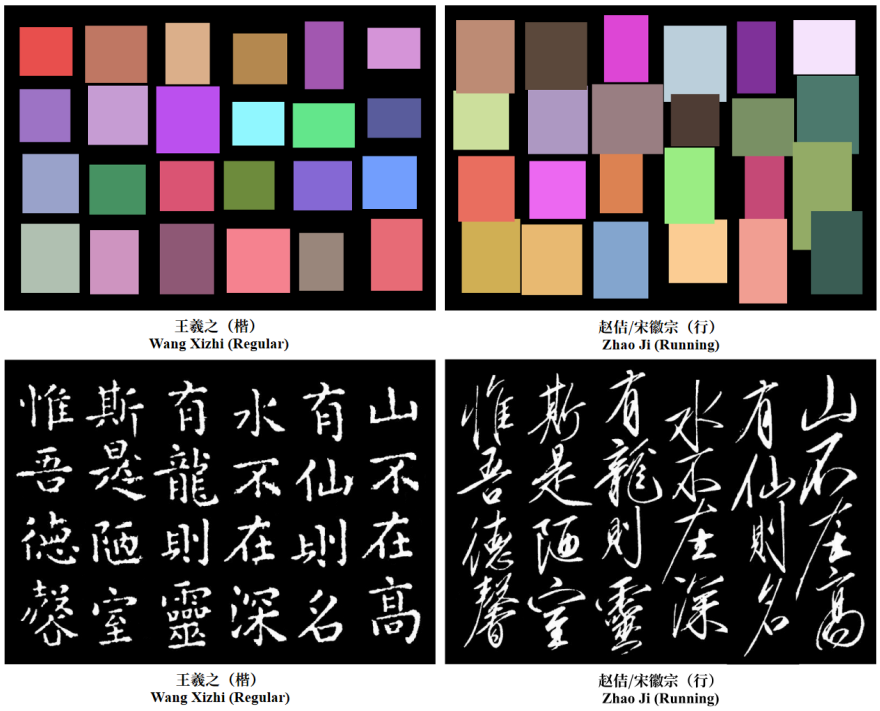



Figure 12: **Wang Xizhi (regular) and Zhao Ji (running). Top:** spatial layout for calligraphers' styles, showing another set of compositions under different style descriptions. **Bottom:** the corresponding generated images, showing the corresponding brushwork and page composition.

### E.3 Additional Generated Samples

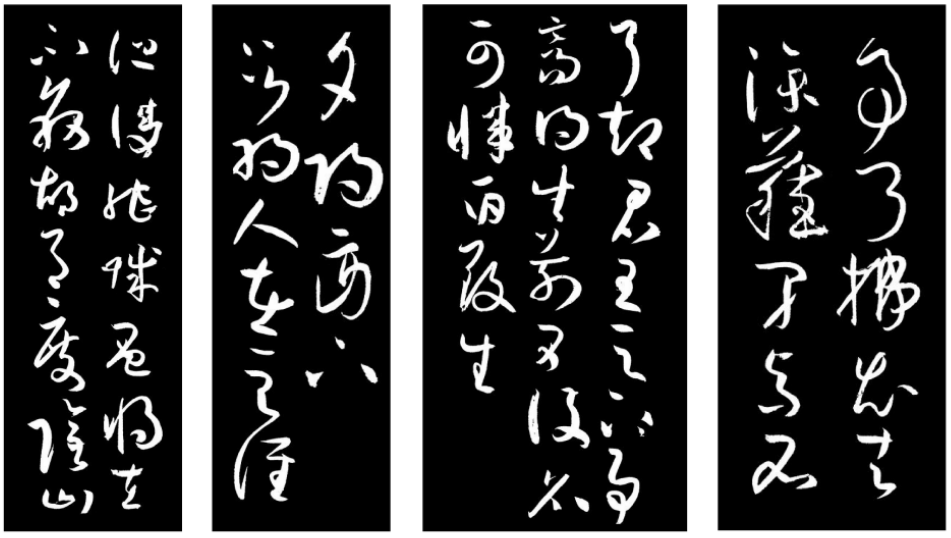
Figures 13–18 present an extended gallery of CalliMaster outputs spanning a wide range of calligraphers, script types, and textual contents. Each page pairs a different target calligrapher with varied phrases drawn from classical and contemporary Chinese literature, illustrating combinations of style and content beyond the main-paper examples.

Calligraphic styles vary in visible ink texture and character geometry as well as page composition. The gallery illustrates how CalliMaster predicts layouts and renders pages under different style descriptions. It is qualitative evidence of variation across the shown examples; a broader controlled evaluation would be needed to establish generalization or unchanged glyph accuracy across styles.



王羲之（行）
Wang Xizhi (Running)

Figure 13: Additional generated samples (page 1 of 6). This collection shows various page-level calligraphy works generated in the style of Wang Xizhi.



孙过庭（草）
Sun Guoting (Cursive)

Figure 14: Additional generated samples (page 2 of 6). This collection shows various page-level calligraphy works generated in the style of Sun Guoting.




乾隆（楷）
Qian Long (Regular)

Figure 15: Additional generated samples (page 3 of 6). This collection shows various page-level calligraphy works generated in the style of Qianlong.

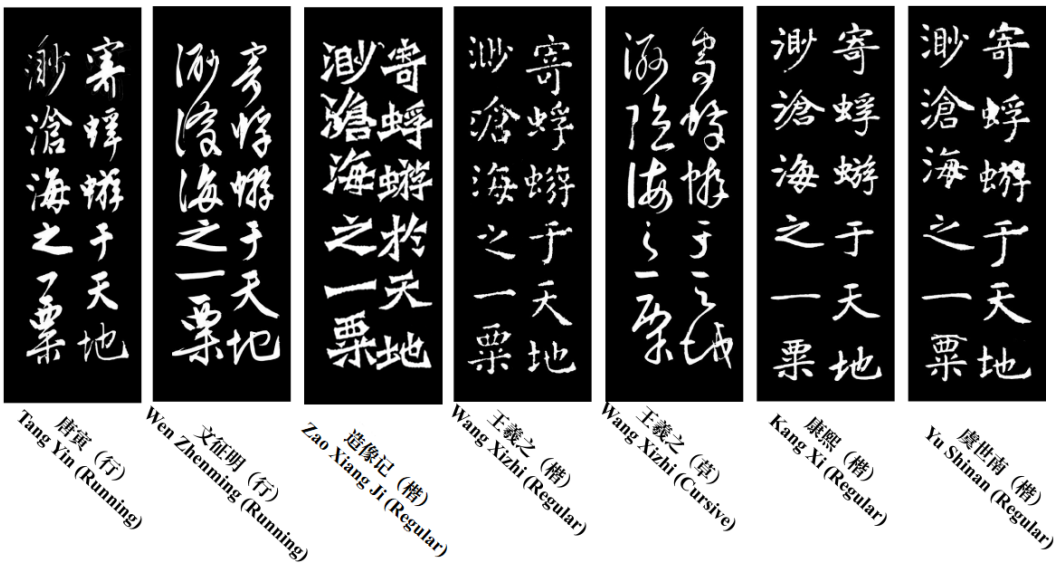



Figure 16: Additional generated samples (page 4 of 6). This collection shows a comparison of different calligraphers and script styles for the same poem.

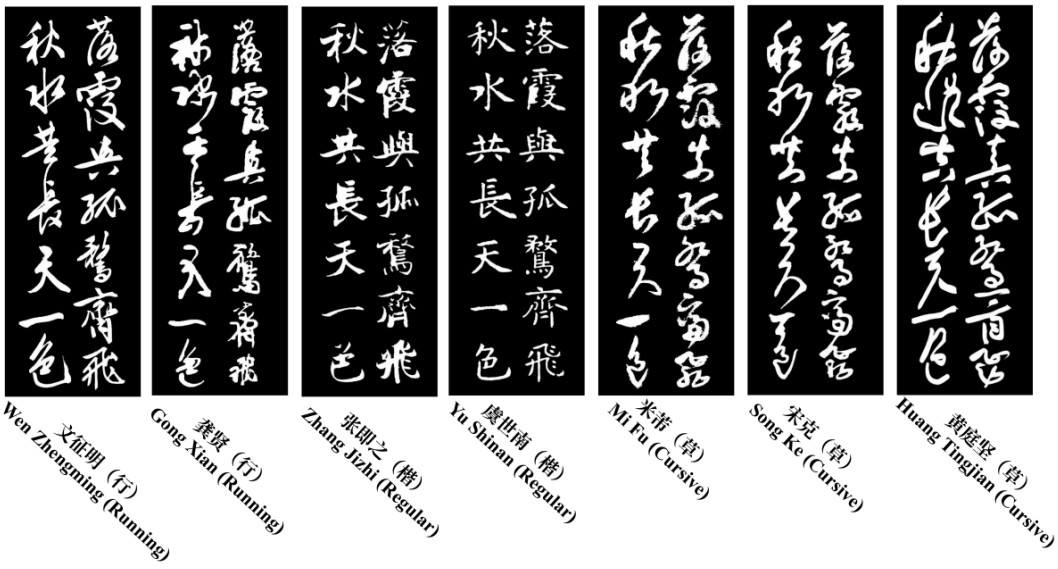



Figure 17: Additional generated samples (page 5 of 6). This collection shows a comparison of various calligraphers and script styles for the same poem.

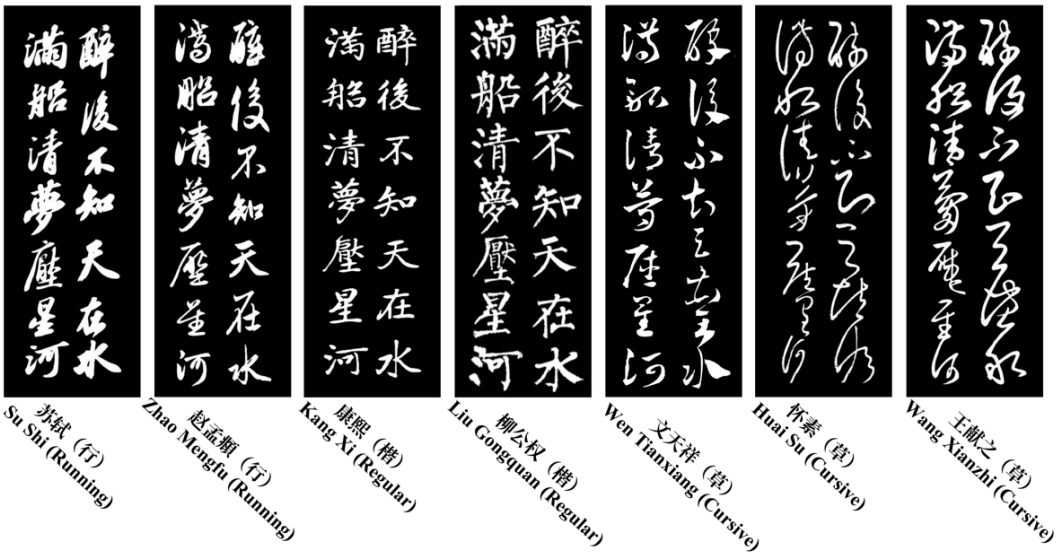



Figure 18: Additional generated samples (page 6 of 6). This final collection shows more examples of different calligraphers and script styles for the same poem.

## F Training-Free Applications of the Joint Regime

The two downstream applications described below reuse the trained network without task-specific fine-tuning. Restoration uses observation replacement during sampling, while DRS uses single-step reconstruction evaluations. Both are consequences of the Joint training regime, in which layout and image are denoised at a shared noise level: that shared level is what allows the sampler to be run with part of its input clamped to observed content (Sec. F.1), and what makes the learned velocity field a well-defined descriptor of how closely the model reconstructs a page it is shown (Sec. F.2). We report both as feasibility studies. The sample sizes are small and no expert-annotated ground truth exists for either task, so neither is offered as a benchmark result.

### F.1 ARTIFACT RESTORATION

**Constrained Joint Inference.** We treat restoration as a constrained joint inference problem. Given a damaged image $\mathbf{I}_{\text{corr}}$ and a binary mask $\mathbf{M}$ (where $M_{ij} = 1$ denotes missing regions), we recover the latent $\mathbf{z} = [\mathbf{x}, \mathbf{b}]$ inside the missing regions while keeping it in harmony with what survives. The Joint training regime is what makes this possible: it is the only regime in which layout and image are denoised at a shared level $t_{\text{img}} = t_{\text{box}} = t$, so both streams can be evolved together. At each integration step $t_i \to t_{i-1}$ we re-impose consistency with the uncorrupted regions using the *Replacement Method* (Lugmayr et al., 2022):

$$\mathbf{z}_{t_{i-1}} = \mathbf{M} \odot \hat{\mathbf{z}}^{\text{model}}_{t_{i-1}} + (1 - \mathbf{M}) \odot \underbrace{\big((1 - t_{i-1})\mathbf{z}_0^{\text{GT}} + t_{i-1}\boldsymbol{\epsilon}\big)}_{\text{noisy observation}}, \tag{15}$$

where $\hat{\mathbf{z}}^{\text{model}}_{t_{i-1}}$ is the ODE prediction and $\mathbf{z}_0^{\text{GT}}$ is the encoder projection of $\mathbf{I}_{\text{corr}}$. Replacement enforces consistency at observed latent locations while the model estimates missing content. Exact preservation of unmasked pixels and seamless boundaries are not guaranteed by latent-space replacement.

**Evaluation.** Due to the lack of paired real-world datasets for calligraphy restoration, we simulate damage by masking character regions from pristine images. We evaluate joint inpainting on 10 such samples. The mean LPIPS is 0.1153 and CLIP Score is 0.9523 (Fig. 19 and Tab. 10). This small experiment illustrates feasibility under the specified masks; it does not establish performance on physical damage.

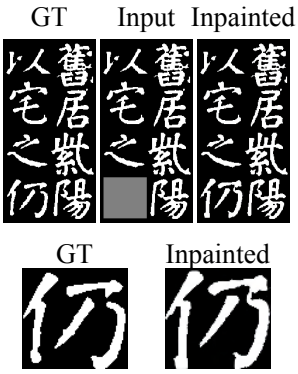



Figure 19: Inpainting example.

Table 10: Inpainting quality on 10 masked regions from real calligraphy images.

| Sample | LPIPS ↓ | CLIP Score ↑ |
|---|---|---|
| 0 | 0.0942 | 0.9650 |
| 1 | 0.1758 | 0.9169 |
| 2 | 0.1460 | 0.9709 |
| 3 | 0.0330 | 0.9806 |
| 4 | 0.1461 | 0.9381 |
| 5 | 0.0998 | 0.9689 |
| 6 | 0.2485 | 0.8295 |
| 7 | 0.0525 | 0.9842 |
| 8 | 0.0651 | 0.9871 |
| 9 | 0.0918 | 0.9822 |
| **Mean** | **0.1153** | **0.9523** |

### F.2 AUTHENTICITY ANALYSIS VIA THE DIFFUSION RECONSTRUCTION SCORE

**Definition.** The Diffusion Reconstruction Score (DRS) summarizes how well the trained velocity field reconstructs a query page under noise. Different noise levels may emphasize different visual scales, but this interpretation has not been independently validated. Given an encoded query image $\mathbf{x}_{\text{query}}$ and a set of noise levels $\mathcal{T} = \{t_1, \ldots, t_K\}$ within $(0, 1)$, we corrupt its latent to $\mathbf{x}_{t_k}$ at each level and take a single flow step back:

$$\hat{\mathbf{x}}_0^{(t_k)} = \mathbf{x}_{t_k} - t_k \cdot v_\theta(\mathbf{x}_{t_k}, t_k, \mathbf{c}, \mathbf{b}), \tag{16}$$

where $v_\theta$ is the predicted velocity. Under Eq. (1), the reconstruction MSE equals $t_k^2$ times the velocity-prediction MSE for that noisy sample. Raw error growth across noise levels therefore also reflects this explicit scale factor. The score is the weighted ensemble of the resulting reconstruction errors:

$$\mathcal{S}(\mathbf{x}_{\text{query}}) = \sum_{k=1}^{K} w_k \cdot \text{MSE}(\mathbf{x}_{\text{query}}, \hat{\mathbf{x}}_0^{(t_k)}). \tag{17}$$

A low score means that the model reconstructs this page well under the specified conditioning. It does not by itself establish that the page is authentic or locate a particular defect.

**Protocol.** We evaluate DRS at nine uniformly spaced noise levels ($t = 0.1$ to $0.9$). We use the following visual-scale interpretation as a working hypothesis:

- **Low noise scales** ($t = 0.1$–$0.3$) may be sensitive to stroke edges, connections, and ink texture.
- **Mid noise scales** ($t = 0.4$–$0.6$) may respond to character-level structure.
- **High noise scales** ($t = 0.7$–$0.9$) may respond to broader composition.

Without expert-labeled authentic–counterfeit pairs in this evaluation, we compare outputs from different generation methods and apply controlled perturbations (randomly placed elliptical blobs, L: 8 and M: 20) to real pages. DRS is computed as the cumulative single-step reconstruction MSE over nine noise levels with three trials each.

**Results.** The main paper reports the aggregate scores in Tab. 4 and their variation across noise levels in Fig. 5. The rise under blob perturbations demonstrates sensitivity to those changes but does not identify which visual property drives each difference.

We stress the limits of this analysis. The perturbation study establishes only that DRS responds to degradation of real calligraphy, and the cross-method comparison necessarily rewards proximity to *our* model's learned distribution, which is not the same thing as authenticity. Validating DRS as an authentication tool would require expert-annotated authentic–counterfeit pairs, which are not included in this study; we therefore present it as an exploratory instrument.