CalliReader識: Contextualizing Chinese Calligraphy via an Embedding-Aligned Vision-Language Model

Supplementary Material

A. Overview

Tables 2, 3, 4, and 7 demonstrate that *CalliReader* outperforms existing VLMs and reasoning models in Chinese Calligraphy Contextualization (CC^2). This supplementary material provides additional quantitative and qualitative results, details on modules and datasets, and key findings on *CalliAlign*, highlighting the substantial advantages introduced by our method.

The outline is structured as follows:

- [Sec. B] describes our user study, demonstrating the difficulty of CC^2.
- [Sec. C] evaluates *CalliReader* against conventional, **fine-tuned** OCR tools.
- [Sec. D] explains key modules: YOLO, OrderFormer, and *CalliAlign*.
- [Sec. E] visualizes *CalliReader* mitigating *CalliAlign*'s errors, proving the necessity of integrating both plug-ins and e-IT.
- [Sec. F] details the dataset, CalliBench, and the prompting for LLM-as-a-judge in contextual VQA.
- [Sec. G] presents additional visualizations on CalliBench and general OCR tasks.

B. User Study

To demonstrate the difficulty of recognizing and comprehending Chinese calligraphy, we carried out a user study involving native Chinese speakers. A total of 142 volunteers, spanning diverse age groups and educational backgrounds, were randomly selected to participate. Among them, 18 individuals had prior expertise in calligraphy, while the remaining 124 did not. All of the participants have a high school degree or above. From our CalliBench dataset, 30 questions were randomly selected, encompassing various levels of difficulty. We tasked the volunteers with recognizing all the words written on each page, aligning the setting with full-page recognition for comparative analysis. We also assessed the performance of *CallReader* on the same set of questions.

Figure S1 illustrates the challenges of reading Chinese calligraphy, even for native speakers. Its cursive scribbled writing and diverse layouts have challenged even the experts, showcasing low F1 scores and high edit distance. Contrarily, *CalliReader* surpasses human behaviors with a more than 40% performance gain in F1(0.918 v.s. 0.512) and 50% reduction in NED (0.092 v.s. 0.590) in comparison to expert behaviors. This underscores its potential value in promoting and popularizing the art of Chinese calligraphy,

This demonstrates the strong capabilities of *CalliReader* in calligraphy tasks and its potential application value in the promotion and popularization of Chinese calligraphy.

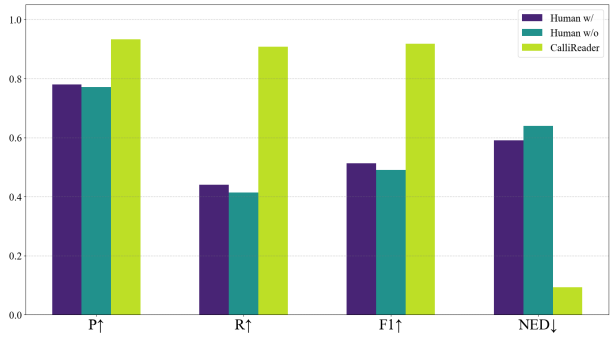


Figure S1. User study and comparison with *CalliReader*. Those with a certain calligraphy background (Human w/) slightly perform better than those without (Human w/o), and both are significantly surpassed by *CalliReader*.

C. Comparisons with OCR Models

CC^2 relies on precise recognition. This section compares *CalliReader* with fine-tuned OCR models on page-level recognition, showing that **simply fine-tuning OCR tools fails to handle scribbled writing and complex layouts in Chinese calligraphy**.

We fine-tuned PP-OCRv4 [9] and EasyOCR [4] on the page-level dataset and evaluated them on CalliBench (hard tier) using F1 and NED. Table S1 shows that *CalliReader* outperforms both, with PP-OCRv4 achieving only 29.3% F1 and a high NED, indicating severe word order confusion. Figure S2 further illustrates these limitations, where fine-tuned OCR models produce completely irrelevant outputs, unable to handle complex calligraphic forms.

Conventional OCR introduces too many inductive biases and requires a large amount of data for training, thus making it unsuitable for calligraphy. In contrast, *CalliReader* integrates character-wise slicing and *CalliAlign* to generate pseudo-text embeddings, enabling the LLM to cross-reference pseudo-text embeddings with image tokens from the ViT encoder. This hierarchical processing refines visual understanding, mitigates errors, and significantly enhances accuracy. Our pluggable slicing and alignment modules further optimize visual token processing and semantic representation, improving recognition without compromising generalization.

Results in Table S3 and S4 verify CalliReader's generalization and robustness. It achieves improved average per-

Figure S2. *CalliReader* can identify scribbled writing while InternVL2 hallucinates, and fine-tuned PP-OCR and EasyOCR fails.

Model	F1↑	NED↓
CalliReader	**0.61**	**0.51**
PP-OCR+ft	0.29	0.94
EasyOCR+ft	0.06	0.98

Table S1. Comparison between *CalliReader* and fine-tuned OCR models on full-page, hard tier.

Dataset	IoU ↑	P ↑	R ↑	F1 ↑
Easy	0.926	0.981	0.995	0.988
Medium	0.929	0.976	0.993	0.984
Hard	0.898	0.978	0.830	0.898
MTHv2	0.802	0.961	0.972	0.967

Table S2. YOLO bounding-box detection results on all tiers and MTHv2 dataset.

Method	Recognition	Extraction	Parsing	Understanding	Reasoning	Average
InternVL2-8B	20.6	**45.2**	23.2	**54.4**	**38.1**	36.3
CalliReader	**58.3** (↑ 183%)	39.2	**27.7**	47.3	34.1	**41.3**

Table S3. Performance of CalliReader on OCRBench v2 (CN).

Method	Recognition	Referring	Spotting	Extraction	Parsing	Calculation	Understanding	Reasoning	Average
InternVL2-8B	49.9	23.1	**0.5**	**65.2**	24.8	26.7	**73.5**	**52.9**	**39.6**
CalliReader	**54.9**	**25.1**	0.1	46.6	**26.2**	**30.7**	72.8	52.6	38.6

Table S4. Performance of CalliReader on OCRBench v2 (EN).

formance on Chinese and is comparable to the base model on English tasks.

D. Model Details

D.1. YOLOv10 for Bounding-box Detection

YOLO (You Only Look Once) is a lightweight, **versatile** object detection model originally designed for **real-time** detection. In character-wise slicing, YOLOv10 [13], is applied for fast and effective character bounding-box detection with a single label (0 for box) for simplicity.

Trained on our page-level dataset, YOLO achieves high-precision bounding box detection. Table S2 reports its IoU, precision, recall, and F1 scores across easy (structured), medium, and hard (cursive, chaotic) layouts, highlighting its adaptability. By segmenting text regions effectively, YOLO reduces page-level CC^2 to sequential recognition and interpretation. On the unseen MTHv2 benchmark, it achieves an F1 of 0.967 and an IoU of 0.802, further demonstrating its robustness. This generalization builds a profound recognition foundation for *CalliReader*, enhancing its accuracy and boosting downstream reasoning abilities.

We further compare the bounding box detection speed of the YOLOv10 and OCR models. As shown in Table S5, YOLO achieves the highest FPS due to its real-time efficiency. Its accuracy and speed ensure that our plug-and-play modules do not significantly impact VLM inference speed or introduce substantial computational overhead.

D.2. OrderFormer: Layout-Aware Sorting

This section details the design and training of *OrderFormer*.

D.2.1. Architecture

Calligraphy layouts, though intricate, adhere to atomic human writing conventions, such as columnar reading order. However, higher-level writing rules remain difficult to formalize due to the fluid nature of calligraphic composition. To address this, we propose *OrderFormer*, a lightweight sorting module with only 0.01B parameters. This four-layer transformer encoder reorders columns into the correct sequence, constraining sequence length to a maximum of 50.

YOLO-detected boxes first undergo the following preprocessing steps:

1. **Clustering** groups vertical columns on spacing and character sizes, distinguishing content from signatures.
2. **Re-scaling** normalizes box coordinates to the top-left origin and scales by image dimensions (W, H), ensuring numerical stability while preserving layout integrity.
3. **Pre-sorting** standardizes to approximate the reading sequence, enhancing training efficiency.

The processed input forms a tensor of shape (B, N, d), where B is the batch size, $N = 50$ is the maximum sequence length, and $d = 4$ represents normalized bounding box coordinates. The output tensor $(B, N, 1)$ provides the sorted indices for bounding boxes.

Given an input sequence $(B_1, B_2, \ldots, B_n)$, the model learns a mapping f such that:

$$f((B_1, B_2, \ldots, B_n)) = (id_1, id_2, \ldots, id_n), \qquad (1)$$

where id_j means the reading order of the j-th box.

D.2.2. Training and Inference

We generate 57,627 column-order samples with diverse layouts for training. The model minimizes MSELoss $\mathcal{L}order$ to learn the correct reading order. We uses AdamW (lr=2×10^{-4}, weight decay=0, amsgrad) with a CosineAnnealingWarmRestarts scheduler ($T_0 = 10$, $T\text{mult} = 2$, $\eta_{\min} = 1 \times 10^{-6}$). Shorter sequences are padded with $[0, 0, 0, 0]$. The model trains for 1000 epochs with a batch size of 4, ensuring robust layout-to-order mapping.

During inference, padding tokens are removed, and each output value is mapped to its order. For example, given output $[2.1, 0.3, 1.2, 4.4, 0.1, -0.1]$ and an original sequence of 4 boxes, the result is $[2, 0, 1, 3]$. This fault-tolerant design preserves order despite minor output variations.

D.3. *CalliAlign* for Character-wise Alignment

CalliAlign transforms single-character images into pseudo-text embeddings, reducing computation by 98.8%, from 256 tokens for each image to just 3 text tokens for each character. This enables efficient recognition of long calligraphic scrolls with over 500 characters. An alternative approach is encoding s characters in one sliced image and conducting alignment, but this introduces several challenges:

Figure S3. Visualizations of LLM mitigating misalignment. Compared to directly decoding *CalliAlign*, *CalliReader* shows better compatibility with the combination of visual tokens and pseudo-text embeddings.

Model	IoU ↑	FPS ↑
YOLO	**0.898**	**11.1**
PP-OCR+ft	0.774	4.1
OpenOCR	0.390	5.4
EasyOCR+ft	0.163	1.1

Table S5. Detection accuracy and efficiency on full-page, hard tier. Our YOLO slicing achieves the highest FPS and IoU, introducing precise visual content to *CalliReader* with less time complexity.

- **Mapping Ambiguity.** Simultaneously aligning batched images with their semantics is hard due to unclear many-to-many mappings, especially with varied sizes.
- **Redundant Spatial Information.** Batching may destroy the reading order and complicate training. Our method preserves the original position information [6].
- **Loss of Characters.** Grouping, like multi-slicing, leads to character omissions, reducing recall.

We have ablated the training of *CalliAlign* by adding other losses, including ratio loss $\mathcal{L}_{rat}$ and contrastive distillation loss $\mathcal{L}_{crd}$, which can be formulated as

$$\mathcal{L}_{rat} = w \cdot \frac{1}{N}\sum_{i=1}^{N}\left(\frac{|y_i - \hat{y}_i|}{|y_i| + eps}\right) + \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2).$$

$$\mathcal{L}_{crd} = \sum_{i\in I}\frac{-1}{|P(i)|}\sum_{p\in P(i)}\log\frac{\exp(z_i \cdot z_p/\tau)}{\sum_{a\in A(i)}\exp(z_i \cdot z_a/\tau)}. \tag{2}$$

Our ablation results demonstrate that these additional losses actually degrade the performance of *CalliAlign*, and therefore, we choose not to use them.

E. LLM Mitigates Misalignment

While pluggable modules like *CalliAlign* offer initial promise for projecting visual characters to their textual embeddings, standalone deployment risks cascading failures. Directly appending character-wise slicing and *CalliAlign*

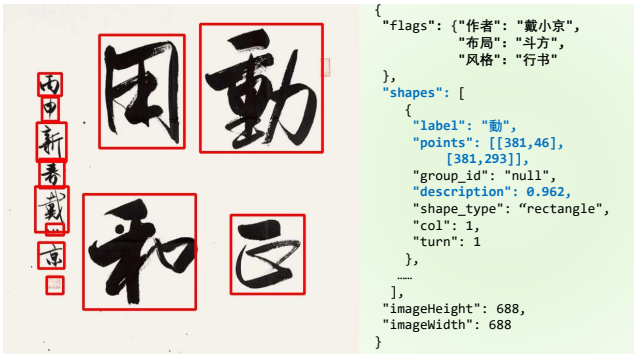
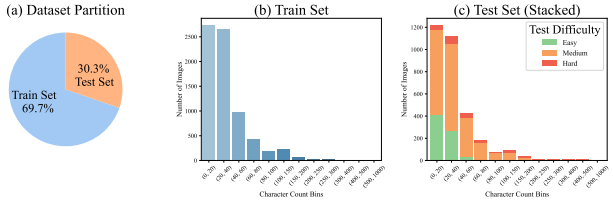


Figure S4. Annotation Format. Left: A piece of Chinese calligraphy. Right: We use the LabelMe format for annotation, recording authority, layout, and style in the *flags* field, while the correct reading order is documented in *row* and *column*.

before and after ViTs frames an OCR-like model. Such OCR-like behavior fails CC^2 contextualization tasks (e.g., linking cursive glyphs to Tang-dynasty poetry allusions), and is prone to erroneous identification, already quantified in paper Table 6 (row2 v.s. row3). This section provides visualized evidence, suggesting *CalliReader* gains from the refining ability of e-IT fine-tuned LLM.

For calligraphic images, we calculated the cosine similarity $\mathcal{C}$ between each pseudo-text embedding from *CalliAlign* and the original embedding table, identifying the nearest neighbor ID as the corresponding token for decoding. This provides a preliminary performance estimation for using *CalliAlign*. We also compared these results with *CalliReader*'s direct outputs.

As illustrated in Figure S3, the direct decoding of *CalliAlign*, due to its character-wise slicing approach, successfully preserves the correct reading order. However, many characters exhibit ambiguous alignments with low $\mathcal{C}$ values, leading to decoding errors. In contrast, *CalliReader*, boosted by its e-IT fine-tuned LLM, demonstrates enhanced



(d) Layout statistics in our collected page-level dataset

Type	横幅 Banner	斗方 Squared sheet	册页 Album	条屏 Hanging scroll	中堂 Middle scroll	楹联 Couplets	手卷 Hand scroll
Train	1629	248	221	1564	1363	2182	150
Test	414	40	2138	237	170	177	16

(e) Calligraphy typeface statistics in our collected page-level dataset

Type	篆书 Seal script	隶书 Clerical script	楷书 Regular script	行书 Running script	草书 Cursive script
Train	381	537	1274	2948	2217
Test	51	220	592	1227	1102

Figure S5. Statistics of our annotated page-level calligraphy dataset. We apply the bounding boxes to train YOLOv10 and *OrderFormer*, while content is used for e-IT. The test set is further derived for multi-grain CC^2 evaluation.

performance. The outputs are semantically coherent and have fewer errors, leveraging the inherent capabilities of the LLM to refine and correct the pseudo-text embeddings.

F. Dataset and CalliBench

This section details our page-level calligraphy dataset (10,549 annotated pages) and introduces CalliBench - a novel benchmark diverging from conventional OCR and VQA tasks. Unlike predecessors that address text recognition or scene understanding in isolation, CalliBench uniquely integrates three objectives: (1) Visual content recognition across diverse script styles with hallucination detection, (2) Historical context grounding through multimodal pretraining, and (3) Knowledge-intensive reasoning for joint analysis of linguistic content, artistic style, and compositional semantics.

F.1. Structured Annotation Framework

Curated from ArtronNet [8] and CAOD [1], our page-level dataset features high resolution and diverse styles. Our hierarchical annotation schema extends LabelMe[11] with domain-specific attributes (Fig.S4). Each JSON entry contains such basic information:

- **Metadata**: Author attribution, style, and layout labels stored under *flag*.
- **Geometric Features**: Per-character bounding boxes, column/row indices, and reading-order coordinates in *shapes*.
- **Paleographic Details**: Character-level labels with Unicode mappings, stroke-order variants, and style classifications (seal script → clerical → cursive)

In Fig. S5, the dataset exhibits broad coverage across character numbers (Fig. S5 b), layout diversity (Fig. S5 d), and diverse styles (Fig. S5 e). All annotations undergo cross-validation by annotators, achieving inter-annotator agreement on character segmentation and labeling. For constructing context-oriented benchmarks such as creation motivation and bilingual interpretation, we introduce calligraphy experts to handle the complexity of annotation. We will continue expanding annotations and will open-source the full dataset and the evaluation benchmark in the future.

F.2. Beyond Recognition: Contextual Benchmark

Figure S6 compares our page-level calligraphy dataset with existing OCR and text-centric VQA benchmarks, emphasizing the broader scope of CalliBench beyond recognition and simple reasoning.

Previous benchmarks are either structured for document analysis or focused on sparse text in natural scenes, limiting their ability to support deep reasoning. OCR benchmarks like MTHv2 [7] contain fragmented, printed contexts, missing coherent context, and lack artistic style. SCUT_HCCDoc [15] comprises handwritten text and structured layouts, offering little variation for reasoning. OCR-Bench [5] and TextVQA [12] focus on identifying scattered text in real-world images. The lack of consistent and contextualized content enables only shallow content-related Q&A, which emphasizes text spotting over reasoning. Furthermore, none of these benchmarks stress the hallucination issues in VLM's recognition process.

CalliBench emphasizes precise full-page calligraphic recognition and contextualization. It doesn't challenge the models to inspect every nook and cranny. In contrast, it requires accurate recognition of the entire calligraphic content, while addressing the hallucination issue through regional detection faithfulness. This approach advances the community's understanding of model reliability and introduces a knowledge-intensive evaluation framework. By combining style, layout, authorship grounding, bilingual interpretation, and higher-level intent analysis, CalliBench emphasizes comprehensive historical reasoning over extensive cursive and scribbled written content, a feature lacking in previous assessments.

F.3. Intent Analysis in Contextual VQA

This section specifically details our evaluation approach to CalliBench's Intent Analysis task. Unlike OCR-centric benchmarks where answers can be retrieved through localized text spotting (e.g., "What's written on the shop sign?"), we employed large language models (LLMs) as judges to assess the open-ended responses.

Figure S6. Benchmark comparison: (Left) Scene-text VQA datasets focus on simple visual questions, generally derived from recognition; (Middle) OCR benchmarks are text intensive yet lack visually-reasoned questions; (Right) CalliBench consists of multi-turn visual-text questions at different levels of granularity.

```
[SYSTEM]
As a Calligraphy Intent Analysis Evaluator, assess responses based on core intent alignment. Factual
minutiae are secondary. User input follows JSON format:
{'calligraphy content': "...", 'model_answer': " ..."}

*Step 1: Key Intent Extraction*
Extract Primary Creative Motivation, Intended Usage, Scenario Target Audience from 'calligraphy content'.
*Step 2: Core Assessment*
1. Task Completion (PASS/FAIL). Whether answer is in coherent English and all important intent elements
are mentioned. This directly decides the validity of answer.
2. Intent Accuracy (80% weight). Consider main intent match: 10pts if primary motivation correct (half-
credit for partial matches). Consider context plausibility: 8-10pts for reasonable scenario/audience
interpretation.
3. Basic Support (20% weight). Consider relevant linking: 8-10 points will be awarded for effectively
connecting the content to the intent of the question. Examples are optional but can enhance clarity.
Consider specific analysis: Answers must focus on analyzing the specific content of the given calligraphy
work. Overly broad or generic statements will result in point deductions. There also should be leniency
for minor errors: Small misunderstandings or inaccuracies in highly detailed aspects will be overlooked,
provided they do not significantly impact the overall analysis.
*Step 3: Scoring*
-  FAIL (0pt) only if task completion is totally missing, or primary motivation is completely wrong.
-  Score Calculation:
  (Intent Accuracy × 0.8) + (Basic Support × 0.2)
*Step 4: Output Format*
{
  "basis": {
    "Task Completion": "<Is the answer readable and complete?>",
    "Intent Accuracy": "<Does the answer capture the intent accurately?>",
    "Basic Support": "<Is the answer specific, avoiding overly broad statements? Are minor errors
overlooked if insignificant?>"
  },
  "score": [0-10 score]
}
```

Figure S7. Intent analysis evaluation prompt.

Our evaluation harness uses 500 curated samples, each with dense intent annotations written by experts. The prompt *"What occasion might have inspired the creation of this piece of calligraphy?"* triggers the model to infer upon its recognition. To quantify model performance, we employ DeepSeek-V3 [3] and Qwen2.5-Max [10] for justification. Our scoring prompt encompasses the following aspects for improved justification:

- **Factorized CoT Evaluation.** LLM judges assess response compliance through 4-step verification following Chain-of-Thought (CoT) [14]. This decomposes complex evaluation and improves the rating interpretability.
- **Structured Output Format.** The judge outputs JSON format for structural answering, where it reasons the completeness, accuracy, and supportive evidence for an explainable rating [2].
- **Average Scoring.** We use DeepSeek-V3 and Qwen2.5-Max to mitigate the judge's potential preference. We evaluate each candidate 3 times to ensure a fair rating.

This framework moves beyond OCR-style answer

Model	Type1 (%)	Type2 (%)
CalliReader	**3.6**	**1.2**
InternVL2-8B	4.3	7.5
InternVL2-1B	5.3	17.4

Figure S8. Error types and error proportions across models.

matching. Figure S7 illustrates the structured judgment workflow and dimensional weightings.

F.4. Error Analysis

We conduct a detailed error analysis on CalliBench to better understand the limitations of current models.

• Hallucination errors, such as incorrect reading order and repeated character generation, are among the most frequent issues. These typically occur in complex layouts or cursive scripts, where spatial relationships between characters are harder to resolve. As illustrated in Figure S8, we visualize representative examples and report the distribution of these errors across different models. To isolate hallucinations from standard recognition errors, we apply a strict threshold-based filtering strategy that separates structural anomalies (e.g., repetition, misordering) from character-level inaccuracies.

• Biases in high-level intent analysis are also observed. Some models tend to oversimplify nuanced historical or cultural contexts, omit key background details, or favor more generic interpretations. These limitations highlight the challenge of contextual understanding in CCR tasks. To address this, we plan to integrate Retrieval-Augmented Generation (RAG) in future iterations of our framework. By grounding the model's generation in external reference materials, RAG can help reduce factual drift and improve historical fidelity in contextual reasoning.

G. More Visulizations

This section visualizes *CalliReader*'s performance on CalliBench and broader OCR and VQA benchmarks (MTHv2 and TextVQA), supplementing numerical analyses in Tables 2,3,4 while proving our method's superiority.

G.1. Text-centric Conversations

CalliReader harnesses extensive pre-trained knowledge to facilitate flexible, multi-turn, calligraphy-contextualized conversations, addressing a wide range of user needs with precision and depth. As demonstrated in Figure S9, *CalliReader* showcases its exceptional versatility in handling calligraphy interpretation across diverse styles and layouts.

Take, for instance, the interaction involving a squared-sheet calligraphic piece (**top-left in Figure S9**), where *CalliReader* not only accurately identifies and translates the content but also provides in-depth context about the author's background and historical significance. Furthermore, it meticulously explains the script style, revealing the intricate artistic choices behind the piece.

In the **top-right examples**, *CalliReader* demonstrates its ability to recognize artists and interpret their creative motivations with remarkable insight. The left couplet, for example, is revealed to celebrate Mr. Guangtang's birthday, while the right piece commemorates a calligrapher's visit to Fuzhou's scenic spots, offering a glimpse into the artist's journey and inspiration.

Another compelling example is found in the **second column**, where *CalliReader* translates a poem into English and introduces the poet, Li Bai, with rich historical and cultural context. This showcases how *CalliReader* can transform complex literary and artistic works into accessible and meaningful interpretations, bridging language and cultural barriers.

Through these examples, the versatility of *CalliReader* is vividly illustrated, transforming intricate calligraphic masterpieces into engaging and understandable dialogues. It not only interprets the art but also enriches the experience by connecting it to broader historical and cultural contexts. This innovative approach bridges language and cultural divides, making the profound art of calligraphy accessible and engaging to a global audience.

G.2. Visualization on OCR and VQA Benchmarks

To assess the generalizability of *CalliReader*, we visualize its performance on MTHv2[7], TextVQA[12], OCRBench [5]. These datasets represent distinct challenges in text-centric tasks: MTHv2 focuses on high-density, small-font historical texts; TextVQA emphasizes real-world scene text understanding; and OCRBench targets robust text spotting across diverse layouts.

As shown in Figure S10, *CalliReader* successfully deciphers extensive, contextually rich historical documents, achieving precise character-level recognition. Our method also extends to text-centric VQA tasks, as evidenced by experiments on TextVQA [12] and OCRBench [5]. These datasets test the ability to understand and reason about text in real-world images and diverse layouts. While our primary focus lies in calligraphy analysis, *CalliReader* demonstrates robust performance across these benchmarks, highlighting its generalizability to broader OCR and VQA tasks.

G.3. More CalliBench Results

G.3.1. Full-page Recognition on Diverse Layouts

Figure S11 showcases the full-page recognition capabilities of *CalliReader* across calligraphic styles and layouts. Our collected page-level dataset encompasses seven primary layout types: banners, squared sheets, calligraphy albums, hanging scrolls, middle scrolls, couplets, and hand scrolls. These layouts feature diverse image aspect ratios, a wide range of calligraphic styles (from seal script to cursive writing), and complex backgrounds with varying colors and patterns.

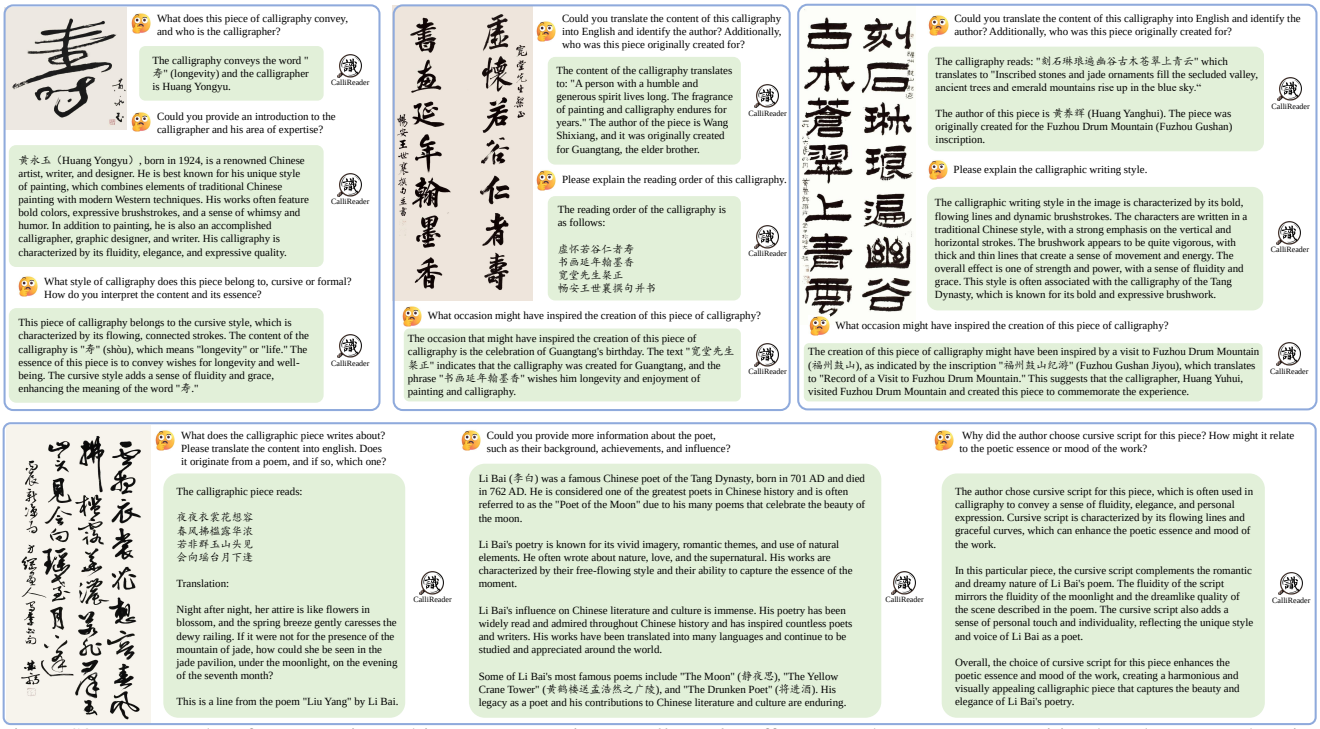


Figure S9. More results of text-centric, multi-turn conversations. *CalliReader* offers not only accurate recognition but also comprehensive insights into calligraphy, including an understanding of its intricate creative background and aesthetic significance.

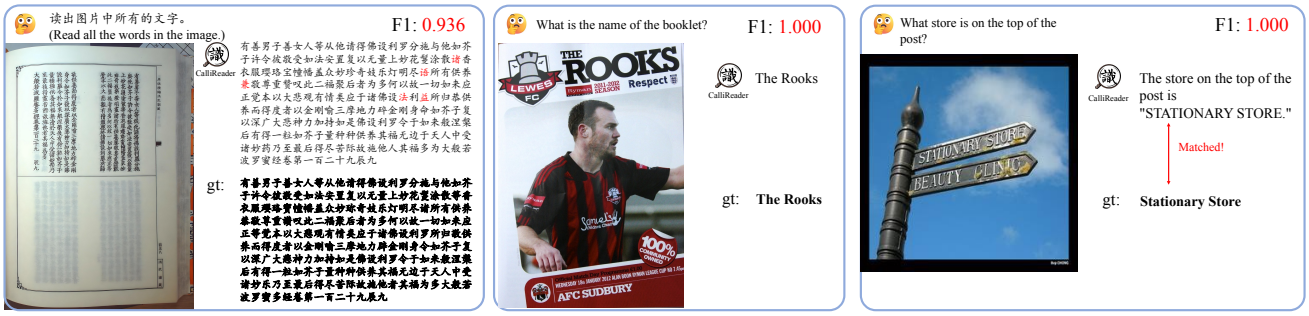


Figure S10. Recognition results on MTHv2, TextVQA, and OCRBench. *CalliReader* has the potential to generalize to other visual text recognition and understanding tasks.

For example, couplets are usually read from right to left. However, as shown in the third example of the fourth row in Figure S11, readers should read the main content from right to left and then read the signature in the same order. This variability often renders rule-based methods for detecting reading orders ineffective. However, *CalliReader* leverages the semantic understanding capabilities of VLMs to autonomously determine the correct reading direction, accurately restoring the intended textual content.

Additionally, *CalliReader* demonstrates robustness in handling other challenges, such as intricate backgrounds (e.g., calligraphy albums, **row 2**) and tiny characters (e.g., square sheet, **row 2**). Furthermore, it exhibits strong recognition performance for cursive writings, as evidenced by comparisons with ground truth, proving its adaptability to diverse and demanding calligraphic scenarios.

G.3.2. Regional Hallucination Detection and Multiple-choice Question

Figure S12 presents additional visual results on regional hallucination detection and multiple-choice questions. VLMs fine-tuned with image-text pairs often exhibit significant hallucination effects when handling incomplete text, frequently producing irrelevant phrases or repeating content. This aligns with our hypothesis that VLMs rely heavily on memorization and guesswork when performing textual recognition.

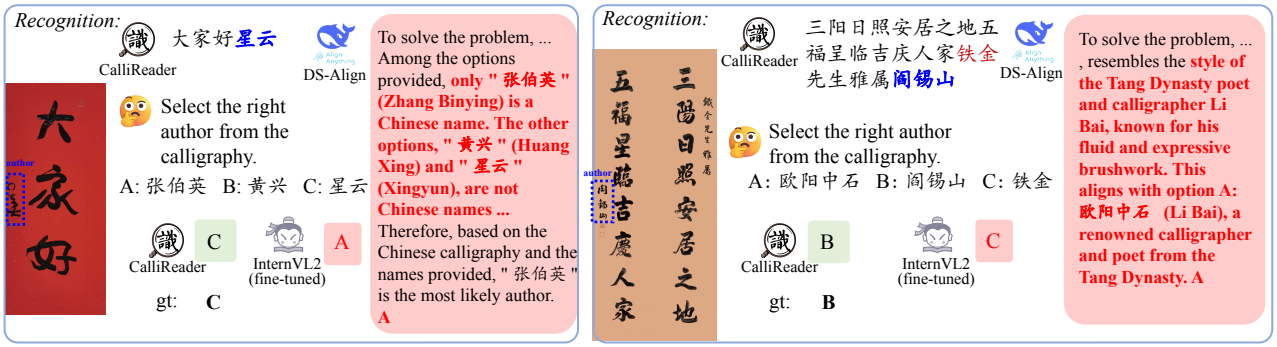
In contrast, the e-IT approach unifies all inputs into a

Figure S11. More full-page recognition results on diverse styles and layouts.

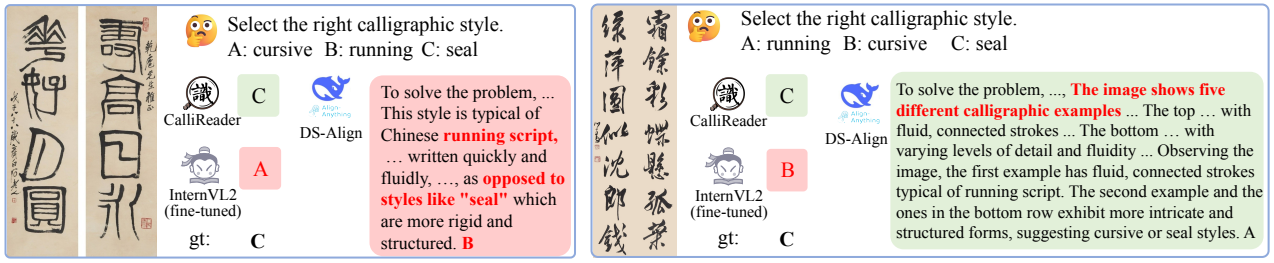
shared embedding space, effectively enabling the model to utilize pseudo-text embeddings during inference. This improves recognition accuracy and mitigates hallucination effects in complex Chinese calligraphy recognition tasks.

The character-wise slicing strategy further enhances CC^2 at varying scales, allowing *CalliReader* to accurately recognize small inscription details in calligraphy artworks, such as signatures and annotations. Performance improvements in authority can be attributed to the inclusion of pseudo-text embeddings that may provide the name of the author, while enhancements in style and layout recognition are likely derived from the integration of visual cues in the calligraphy content. This enables the model to draw upon prior knowledge to provide a deeper understanding of calligraphy works.

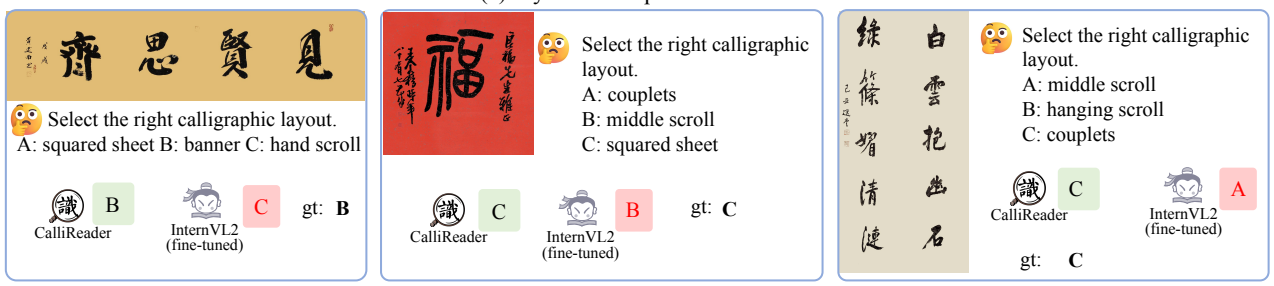
(a) Region hallucination detection.



(b) Author choice questions.



(c) Style choice questions



(d) Layout choice questions

Figure S12. More results on regional hallucination detection and knowledge selection.

References

[1] CAOD. Chinese art open data, 2024. Accessed: 2024-10-12. 4

[2] David Chan, Suzanne Petryk, Joseph Gonzalez, Trevor Darrell, and John Canny. CLAIR: Evaluating image captions with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13638–13646, Singapore, 2023. Association for Computational Linguistics. 5

[3] DeepSeek-AI. Deepseek-v3 technical report, 2024. 5

[4] JaidedAI. Easyocr, 2024. Accessed: 2024-10-25. 1

[5] Yuliang Liu, Zhang Li, Biao Yang, et al. On the hidden mystery of ocr in large multimodal models. *arXiv preprint arXiv:2305.07895*, 2023. 4, 6

[6] Jinghui Lu, Haiyang Yu, Yanjie Wang, et al. A bounding box is worth one token: Interleaving layout and text in a large

language model for document understanding. *arXiv preprint arXiv:2407.01976*, 2024. 3

[7] Weihong Ma, Hesuo Zhang, Lianwen Jin, et al. Joint layout analysis, character detection and recognition for historical document digitization. *ICFHR 2020*, 2020. 4, 6

[8] Artron Net. Artron net - art searching engine, 2024. Accessed: 2024-10-12. 4

[9] PaddlePaddle. Paddleocr, 2024. Accessed: 2024-10-25. 1

[10] Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. 5

[11] Bryan C Russell, Antonio Torralba, Kevin P Murphy, and William T Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77:157–173, 2008. 4

[12] Amanpreet Singh, Vivek Natarjan, Meet Shah, et al. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019. 4, 6

[13] Ao Wang, Hui Chen, Lihao Liu, et al. Yolov10: Real-time end-to-end object detection. *arXiv preprint arXiv:2405.14458*, 2024. 2

[14] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, pages 24824–24837. Curran Associates, Inc., 2022. 5

[15] Hesuo Zhang, Lingyu Liang, and Lianwen Jin. Scut-hccdoc: A new benchmark dataset of handwritten chinese text in unconstrained camera-captured documents. *Pattern Recognition*, page 107559, 2020. 4