Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

LGE-FIng: Learned Glyph Embedding for Image-Conditioned Candidate Reranking in Post-OCR Correction of Chinese Historical Books

Laixin Shen

slx965@163.com

Huangshan University

Fan Yang

Huangshan University

Research Article

Keywords: Character recognition, cultural heritage informatics, document image analysis, glyph embedding, historical documents, metric learning, post-processing, Viterbi decoding

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11255721/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

LGE-FIng: Learned Glyph Embedding for Image-Conditioned Candidate Reranking in Post-OCR Correction of Chinese Historical Books

Laixin Shen and Fan Yang

School of Information Engineering, Huangshan University, Anhui, China

L. Shen is also with the Huizhou Culture Research Center of Anhui Province (e-mail: slx965@163.com); F. Yang (e-mail: fanyang80@163.com).

Copyright (c) 2016 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

***Abstract*—Post-OCR correction of Chinese historical books is bounded by substitution errors, yet the glyph cues that would resolve them are used poorly: hand-crafted descriptors break down across the domain gap between woodblock printing and modern typefaces, and they are keyed to the recognized character rather than to the observed glyph image. LGE-FIng learns a glyph embedding under an ArcFace objective with cross-domain alignment, so that a woodblock glyph block and a multi-font rendered prototype of the same character share one class center. At each position the embedding is queried by the real image block, and its nearest neighbors feed image-conditioned candidates into the corrector's search lattice. On a 60-page book-disjoint test set, the image-queried channel beats an identical channel keyed to the recognition character's rendering by +1.13 to +1.54 percentage points (pp) under four independent weight-mapping calibrations, all agreeing in sign and magnitude. The gain survives a scale-free rank calibration that equalizes the two channels' weight ladders, which rules out a cosine-scale artifact. A layout-aware fusion of both channels reaches +3.66 pp over the bare recognizer (PP-OCRv5) and +3.73 pp with a real-corpus hybrid gallery. A mechanistic probe traces the effect to rank one: the image channel ranks the ground truth first in 19.4% of substitution positions, while the static channel cannot do so by construction. The method trains cheaply, stays OCR-agnostic, and integrates into an existing corrector as an additional candidate source.**

Keywords: **Character recognition, cultural heritage informatics, document image analysis, glyph embedding, historical documents, metric learning, post-processing, Viterbi decoding.**

1. INTRODUCTION

Industrial-scale digitization of Chinese historical books underpins quantitative work in the humanities, archival search, and cultural-heritage informatics [1], [2]. The gap between modern printed text and woodblock printing, however, limits the accuracy of general-purpose recognizers. Historical pages are set in vertical columns of traditional characters, bounded by ruled frames and interspersed with ornaments and interlinear annotations; engines trained on modern horizontal simplified text degrade sharply on this material [3]–[6]. Because one misrecognized character can change the meaning of a classical sentence, the tolerated error is far lower than in general document analysis.

Two difficulties recur. The first is data: page-level ground truth aligned with images is scarce [1], [7], so many reported results rest on small or non-public samples, as the post-OCR correction shared tasks also observe [8], [9]. The second is attribution: recognizers are usually evaluated as monolithic systems, which leaves the contribution of individual stages unknown. A prior component-wise study on a large character-aligned corpus of this material shows that correction alone is bounded at 92.86% character accuracy, and that substitution, not coverage, is the binding error class [10], [11]. That result reframes the problem: the remaining errors are visually confusable characters that a language model cannot disambiguate without shape evidence.

Shape evidence is exactly what existing glyph cues use poorly. A prior histogram-of-oriented-gradients (HOG) fingerprint [12] injects visually similar candidates and yields a measurable gain, but it has two structural weaknesses. First, it is queried by the recognized character: the candidate set is built from the recognition result's rendering, so the corrector never sees the actual glyph image. Second, the descriptor is defeated by the domain gap: under image-conditioned retrieval its top-1 falls to 0.20%, because modern-font renderings of a character are poor proxies for its woodblock form [11], [12]. Both weaknesses have the same fix: learn an embedding from authentic glyph blocks, and query it with the observed image. Pretrained classical-Chinese language models such as SikuBERT [13] supply in-domain context, but they cannot disambiguate visually confusable characters without shape evidence.

LGE-FIng (Learned Glyph Embedding for post-OCR correction of Chinese historical books) addresses both weaknesses. It learns an embedding with an ArcFace objective [14] in which a real woodblock glyph block and multi-font rendered prototypes of the same character share one class center. This cross-domain alignment lets a rendering-based gallery cover the full vocabulary while the query remains the real image. The embedding then enters the corrector as an image-conditioned candidate channel inside the Viterbi lattice [10], [11], replacing the static, recognition-character-query channel. The setup isolates the contribution: the recognizer, the language model, and the candidate interface are all held fixed, so the only variable is whether the glyph evidence is drawn from the image or from the recognized character.

Our contributions are as follows.

- A learned glyph embedding with cross-domain alignment that raises image-conditioned glyph retrieval top-1 from 0.20% (a HOG baseline defeated by the domain gap) to 74.28% on a rendered gallery, at a cost of only -0.77 pp on the in-domain gallery.
- LGE-FIng, an image-conditioned candidate channel that injects the embedding's nearest neighbors into the corrector's search lattice without changing its decoding or its language model.
- A controlled ablation in which the image-queried and the recognition-character-queried channels share the same encoder and gallery; the only difference is the query vector. The image channel wins by +1.13 to +1.54 pp under four independent calibrations, and the sign holds under a scale-free rank calibration that equalizes the two channels' weight ladders.
- A mechanistic probe over 4,364 substitution positions that localizes the gain to rank one: the image channel ranks the ground truth first in 19.4% of positions, whereas the static channel is structurally prevented from doing so, because its query is the recognized character's own rendering.
- An end-to-end result of +3.66 pp over the bare recognizer with the rendered gallery, rising to +3.73 pp when the gallery is enriched with real corpus instances (book-disjoint, zero-leakage), against a +12.02 pp ceiling set by a recognition-backbone swap.

2. Related Work

2.1. OCR of Chinese Historical Documents

Classical engines such as Tesseract [3] provide a general baseline whose accuracy on woodblock pages is very low. Deep-learning systems including PP-OCR [4] and its PP-OCRv5 successor [15], and TrOCR [5], improve modern-text recognition, and recent work extends recognition to ancient characters with task-specific architectures and data-enhancement strategies [6]. Layout-aware multimodal models such as TongGuOCR [1] report strong results by expanding the tokenizer with rare glyphs and modeling layout jointly, but they assume large-scale image-text supervision and GPU-class training budgets. Another line treats segmentation and reading order as the core difficulty [7], [16]. Our work is recognition-agnostic: it sits downstream of any recognizer and improves the output through shape evidence.

2.2. Post-OCR Correction

Post-processing correctors operate on recognizer output and can be trained from transcriptions alone. Classical approaches combine an error model with a language model through noisy-channel or Viterbi decoding [8], [11]; the ICDAR competition series established the task and the difficulty of repairing real error distributions [8]. Recent variants replace the N-gram model with a pretrained neural or large language model [17], [18], and decoding-time fusion injects language-model scores into the recognizer beam search [19], [20]. The ICDAR 2026 HIPE-OCRepair task [9] confirmed that LLM-assisted correction improves accuracy yet repeatedly over-corrects low-noise input [21]. Our work belongs to the noisy-channel family, but supplies the candidate set with learned shape evidence rather than relying on language context alone.

2.3. Glyph-Aware Recognition and Metric Learning

Dictionary-guided recognition shows that constraining candidates by a language or shape reference resolves visually ambiguous characters [11]. Rendering-based template matching, however, is defeated by the domain gap: a prior cold-start test gave a top-1 of 0.25% against 0.21% for random selection with typeface-rendered templates matched to woodblock forms. Glyph representations must therefore be anchored to authentic samples. Metric learning with an angular-margin objective [14], [22] pulls same-class samples together and pushes different classes apart in a normalized embedding space; cross-domain adaptation aligns representations learned from different sources [23]. We combine both: a shared class center is driven by both real woodblock blocks and multi-font renderings, so the embedding is simultaneously discriminative in-domain and aligned to a renderable gallery.

How such shape evidence enters the corrector matters as much as the representation itself. Confidence-driven designs rescore or route existing hypotheses: recognition scores have been examined as a proxy for OCR quality

[24], decoder uncertainty routes characters between dictionary and free recognition [25], and confidence fused with language-model context localizes errors [26]. All of these signals state that a position is suspicious, yet none describes what the observed glyph looks like, so a confident but wrong character escapes them. Shape-conditioned reranking closes this gap from the opposite side: instead of generating a candidate stream, it attaches a compact, image-conditioned preference term to the corrector's existing search. The present design adopts the reranking role and strengthens it in two ways over template matching: the query is the real glyph image rather than the recognized character's rendering, and the gallery side is learned jointly with the cross-domain alignment above.

2.4. Research Gap

Most reports neither quantify the contribution of individual correction components nor report the operating threshold that keeps a strong baseline intact, and negative ablations are almost never published. Practitioners are left without guidance on where to spend effort. We close that gap with a fully componentized evaluation, an explicit image-versus-character ablation that isolates the query source, and a mechanistic probe showing where the gain comes from.

3. Problem Statement and Preliminaries

3.1. Problem Statement

Let x be a page image and let a recognizer R produce a character sequence $y = R(x)$ of length n. Let y* be the ground-truth transcription of length m. The task is to learn a corrector C that maximizes the character accuracy of C(y) relative to y*, where accuracy is defined by the normalized edit distance

$$A(C(y), y^*) = 1 - d(C(y), y^*) / m \quad (1)$$

with d the Levenshtein distance. The corrector observes y and, optionally, the per-character glyph image. A deployable corrector must not reduce accuracy on pages where y is already accurate:

$$A(C(y), y^*) \geq A(y, y^*) \text{ for all pages} \quad (2)$$

This monotone non-degradation requirement separates a deployable corrector from an aggressive one [9], [21].

3.2. Definitions

Definition 1 (Glyph embedding). A glyph embedding e(I) maps a 64x64 normalized glyph image block I to a unit vector in R^256, trained so that blocks of the same character lie close under the cosine distance regardless of whether the block is a woodblock print or a font rendering.

Definition 2 (Cross-domain alignment). Training pairs a real woodblock block of character c with multi-font rendered prototypes of c in the same ArcFace class center [14], so that a gallery built from renderings covers the full vocabulary while queries remain real images.

Definition 3 (Candidate set). For a recognized character y_i at position i, the candidate set K_i contains y_i, its distributional neighbors in the corpus vocabulary, and the characters whose embeddings are the nearest neighbors of $e(I_i)$. The lattice formed by $\{K_i\}$ is the search space of the corrector, following the dictionary-guided paradigm [11].

Definition 4 (Image and static channels). The image channel queries $e(I_i)$ against the gallery; the static channel queries $e(\text{rendering}(y_i))$, the rendering of the recognized character. Both use the same encoder and gallery, so their difference isolates the query source.

Definition 5 (Context score). The context score of a candidate c at position i is the Witten-Bell-smoothed [10] N-gram probability $P_{ctx}(c \mid y_{i-N+1..i-1})$ estimated from corpus transcriptions, expressed in the log domain.

3.3. Algorithms

Algorithm 1 Offline embedding and gallery construction

```
Input:  aligned corpus {(image block, char)}, font set F
Output: encoder e, gallery G (rendered prototypes + real instances)
for each character c in vocabulary do
    build batch B_c = {real blocks of c} U {render(c, f) : f in F}
    share one ArcFace class center across all samples in B_c
end for
train encoder e by minimizing ArcFace loss over all B_c
G_render <- { e(render(c, f)) : c in V, f in F }   // renderable coverage
G_real <- top-K real instances per char from pool3000 (book-disjoint from eval)
G <- G_render U G_real                            // hybrid gallery
return e, G
```

Algorithm 2 Online image-conditioned correction

```
Input:  recognized text y, glyph blocks {I_i}, encoder e, gallery G, LM P_ctx
Output: corrected text ŷ
y <- OpenCC_simplified_to_traditional(y)  [27]   // script alignment
for i = 1 to n do
    q_i <- e(I_i)                                // image query
```

```
    K_i <- {y_i} U { top-N nearest in G by q_i }
    w(c) <- calibrate(cos(q_i, g_c))              // Section 5.3
end for
score(c, i) <- log w(c) + lambda * log P_ctx(c | context)
ŷ <- Viterbi(argmax over lattice)
return ŷ
```

4. The LGE-FIng Method

LGE-FIng has three stages: (1) train a glyph embedding with cross-domain alignment, (2) build a renderable gallery from the embedding, and (3) inject the embedding's nearest neighbors as an image-conditioned candidate channel into the corrector. Stages 1 and 2 are offline and recognition-agnostic; stage 3 is a drop-in replacement for the static glyph channel.

4.1. Cross-Domain Aligned Glyph Embedding

The encoder is a five-layer fully-convolutional backbone with channels (32, 64, 128, 256, 256), stride-2 convolutions with batch normalization and ReLU, global average pooling, and a 256-dimensional linear projection whose output is L2-normalized. An ArcFace head [14] with angular margin 0.25 and scale 24.0 maps the embedding to class logits. Training uses AdamW (learning rate 3e-3, weight decay 1e-4) with cosine annealing for 40 epochs at batch size 512, and applies random affine augmentation (rotation, scale, shift) to emulate the per-book variation of woodblock printing.

The cross-domain alignment is the central design choice. For each character c, a training batch concatenates the real woodblock blocks of c with multi-font rendered prototypes of c (Song, Kai, Fang, and Hei typefaces), and all samples share one ArcFace class center. This forces the encoder to discard font texture as a shortcut and to bind the woodblock form and the rendered form of c to the same point in the embedding space. Multiple styles are used on purpose: a single-font alignment would overfit the encoder to that font's texture. The same-class-center constraint is standard metric-learning practice [14], [22]; the contribution here is its application to bridging woodblock and rendered glyphs for correction.

4.2. Gallery: Rendered Prototypes and Real Instances

After training, the base gallery is built from rendered prototypes: for every dictionary character c and every training font f, store g_{cf} = e(render(c, f)) and average over fonts. This guarantees full vocabulary coverage (16,274 of 18,383 dictionary characters, 88.5%), and the gallery provides the candidate source for both the image and the static channels; only the query vector differs (Definitions 3-4). Characters that no training font can render are excluded, but because the corrector's vocabulary is a set of common characters this gap is not the current bottleneck.

Because the encoder is cross-domain aligned, authentic woodblock instances of c also lie near g_c in the embedding space. We therefore enrich the base gallery with real instances mined from the 431k-block pool3000 corpus using the same encoder. The 60 evaluation pages are strictly book-disjoint from pool3000, so this enrichment has zero leakage with the test set. For each character we append the K=16 nearest-centroid real instances to the rendered prototype; the hybrid gallery then combines render-based coverage with real-shape neighbors that pull the true character closer at substitution positions. The enrichment changes only the candidate source, not the encoder, the weight mapping, or the decoder.

4.3. Image-Conditioned Candidate Injection

At each position i the image query $q_i = e(I_i)$ retrieves the top-N nearest gallery entries, which are appended to the candidate set with weights derived from the cosine similarity. The weight mapping is calibrated on held-out books (Section 5.3) and maps a similarity s to a candidate weight w = clip(a + b*s, lo, hi). Because the injection point is after the context candidates and after the K-best truncation, the visual evidence is never displaced by weight ordering. The decoder is unchanged: it scores each candidate by log w(c) plus the language-model context score and returns the maximum-scoring sequence by Viterbi decoding [10], [11]. With the image scorer set to None the channel is disabled and the corrector reverts exactly to its prior behavior, which is what makes the ablation controlled.

5. Experimental Setup

5.1. Corpus, Embedding Data, and Ground Truth

Provenance, scope, and availability. The corpus derives from the Dingxiu Ancient Books Full-Text Retrieval Platform [28], a commercial full-text database of Chinese books written and printed from the pre-Qin period to the Republican era, developed by Beijing Hanhai Boya Technology. From the platform's Huizhou-document subset we curate 2,219 volumes, about 130,000 page images of woodblock-printed books, in which every page image is paired

with a per-character transcription in traditional script, so character-level scoring requires no manual annotation. The platform is licensed by institutional subscription (http://www.ding-xiu.com); the derived page-alignment files and the 60-page evaluation bundle are available from the authors for research reproduction.

The embedding is trained on a separate pool of 947,919 glyph blocks from 3,000 pages and 750 books, split book-disjoint into 477 training and 95 test books; after gating, 431,425 blocks cover a vocabulary of 2,952 characters (minimum count 8 and renderable). The gallery covers 16,274 of 18,383 dictionary characters (88.5%). The correction evaluation uses a 60-page set drawn from six volumes, the same held-out pages used in the prior component-wise study, so the results are directly comparable.

5.2. Accuracy Definitions and Reporting

The primary measure is the character accuracy of the ICDAR post-correction task, one minus the Levenshtein distance between prediction and reference divided by the reference length [8], which decomposes into substitutions, deletions, and insertions. We report two aggregations that must not be confused: page-averaged accuracy (mean over pages) and pooled accuracy (all errors divided by the total reference length). The two diverge on this corpus by about 0.78 pp because page lengths vary widely and page-averaging over-weights short pages; following the prior study we report pooled accuracy as the main figure. The substitution-deletion-insertion decomposition is computed by an exact Levenshtein dynamic program, not by hand-parsing difflib opcodes, which mis-attribute aligned edit operations. All deltas are paired per-page differences in percentage points (pp).

5.3. Controls and Calibration

Three controls recur. (i) Every comparison holds the recognizer output byte-identical, so recognition-side variance is constant. (ii) The image and static channels share the same encoder and gallery, so their difference is exactly the query source. (iii) The weight mapping a, b is calibrated on 40,000 queries from held-out books by fitting a monotone function between the cosine similarity and a target weight band [0.20, 0.70] anchored to the prior study's glyph weight range [10]. To rule out a cosine-scale artifact, we also evaluate a scale-free rank calibration that assigns a fixed descending weight ladder to the top-N neighbors regardless of their similarity values; if the image-over-static gain were an artifact of the channels' different cosine scales, it would vanish under this calibration.

5.4. Ablation Arms

Five arms vary only the glyph-prior source: base (no glyph prior), hog_static (the prior HOG fingerprint channel [12]), lge_static (LGE-FIng queried by the recognized character's rendering), lge_image (LGE-FIng queried by the real image block), and lge_both (image channel plus a layout-aware second-orientation channel). The core quantity is lge_image minus lge_static: same encoder, same gallery, same decoder; only the query vector changes.

6. RESULTS

6.1. Retrieval: Learned Embedding vs. Hand-Crafted Descriptor

We first confirm that a learned embedding outperforms the HOG fingerprint under image-conditioned retrieval on the same gallery convention (one vector per class, identical training blocks). Table 1 reports top-1 and top-5 over 1,373 classes with random top-1 at 0.07%.

Table 1
Image-conditioned glyph retrieval: learned embedding vs. HOG (top-1 / top-5)

Configuration	Query	Gallery	top-1	top-5
HOG [12]	image block	real blocks	50.55	68.46
LGE-FIng (ours)	image block	real blocks	73.56	81.95
HOG [12]	image block	font render	0.20	0.70
LGE-FIng (ours)	image block	font render	0.62	1.92

The learned embedding raises top-1 by +23.0 pp on the in-domain gallery. Both methods collapse on the rendered gallery (0.20% / 0.62% top-1): a font rendering is a poor proxy for a woodblock glyph, so the gallery must be aligned rather than replaced.

The collapse of both methods on the rendered gallery (top-1 of 0.20% for HOG and 0.62% for the image-trained embedding) is the decisive observation: even a strong embedding learned only on real blocks cannot retrieve against a rendered gallery. The gallery must therefore be aligned to the embedding, not merely substituted.

6.2. Cross-Domain Alignment

We compare a tile-only baseline (real blocks only) with the mixed training (real blocks plus three-font renderings sharing class centers). Table 2 reports top-1 against three gallery domains. Mixed training raises the rendered-gallery top-1 from 0.20% to 74.28% (+74.08 pp versus the prior HOG channel), while the in-domain gallery drops by only

0.77 pp (78.22% vs. 78.99%). The alignment is near lossless on seen fonts and transfers to unseen fonts at 43.22% top-1 zero-shot.

Table 2
Cross-domain alignment: mixed training vs. tile-only (top-1)

Gallery domain	tile-only	mixed (ours)	Δ top-1
Real blocks (in-domain)	78.99	78.22	-0.77
Rendered, seen font	0.37	74.28	+73.91
Rendered, unseen Kai	0.86	43.22	+42.36
Rendered, unseen MSJH	0.09	16.46	+16.37

Mixed training converts a useless rendered gallery (0.20% top-1 for the prior channel) into a near in-domain one (74.28%) at a cost of -0.77 pp in-domain, a one-sided gain rather than a trade-off.

6.3. Weight Calibration

On 40,000 held-out-book queries the calibrated mapping separates the true character from negative characters with a median cosine of 0.7411 versus 0.4871 (gap +0.2540), and retrieves the true character in the top-8 for 85.57% of queries at median rank 1. The fitted weights are a = -0.7588, b = 1.9685 in the image domain, and a = -0.4571, b = 1.1571 in the render domain when the static channel is calibrated on its own scale. A self-check reproduces the training report's font:multi top-1 of 0.81640 exactly (versus 0.8164), confirming that no index-space error contaminates the calibration.

Table 3
Calibration on 40,000 held-out-book queries

Quantity	Value
cosine pos. (median)	0.7411
cosine neg. (median)	0.4871
separation gap	+0.2540
top-8 hit rate	85.57%
median rank	1
image-domain a, b	-0.7588, 1.9685
render-domain a, b	-0.4571, 1.1571
self-check font:multi top-1	0.81640

6.4. Main Ablation

Table 4 reports pooled character accuracy for the five arms under four weight-mapping calibrations. The core quantity, lge_image minus lge_static, is positive under all four and ranges from +1.13 to +1.54 pp. The sign is therefore robust to the calibration: it is not an artifact of how similarities are turned into weights.

Table 4
Five-arm ablation, pooled ICDAR accuracy (core Δ = lge_image − lge_static)

Weight mapping	base	hog	lge_st	lge_img	lge_both	Δ
direct (prior 0.20+0.50·cos)	73.18	74.70	74.36	75.77	77.37	+1.41
tuned (image-domain calib.)	73.18	74.70	74.26	75.39	77.12	+1.13
tuned (render-domain calib.)	73.18	74.70	74.08	75.39	77.12	+1.31
rank (scale-free)	73.18	74.70	74.32	75.85	77.54	+1.54

base 0.7318 / hog 0.7470 / lge_static 0.7436 / lge_image 0.7577 / lge_both 0.7737 (direct). Δ ∈ [+1.13, +1.54] pp across all four calibrations. lge_both vs. the bare recognizer (PP-OCRv5, 0.7390 pooled) is +3.47 pp with the rendered gallery, rising to 0.7756 (calib) / 0.7763 (rank), i.e., +3.66 / +3.73 pp, with the real-corpus hybrid gallery of Section 4.2.

Two checks address possible confounds. First, the direct mapping is already amplitude-matched to the prior HOG channel ($\cos \leq 1$ implies $0.20+0.50\cdot\cos \leq 0.70$, exactly the HOG weight band), so the image-over-static gain is not a larger-weight artifact. Second, and decisively, the rank calibration assigns the two channels an identical weight ladder (both span [0.350, 0.700], mean 0.525); if the gain were a cosine-scale difference, it would disappear here. Instead the rank calibration gives the largest Δ (+1.49 pp), opposite to the scale-artifact prediction. We therefore attribute the gain to the candidate set and ordering, not to the weight scale.

For context, lge_image beats base by +2.59 pp and the prior HOG channel by +1.07 pp (the learned embedding exceeds the hand-crafted one under matched amplitude); lge_both reaches +4.19 pp over base, exceeding the sum of the image and HOG gains (+3.66 pp) by +0.53 pp, indicating that the two channels act on disjoint error subsets rather than redundantly.

6.5. Error Decomposition and Length Preservation

Table 5 decomposes the direct configuration by error class. Accuracy rises entirely through substitution: the substitution rate falls from 0.1968 (base) to 0.1547 (lge_both), a -4.21 pp reduction consistent with the +4.19 pp accuracy gain, while deletion and insertion rates are unchanged across arms.

The LGE-FIng retrieval stage is gallery-source agnostic. Beyond the rendered prototype gallery of Section 4.2, we deploy a hybrid gallery that enriches it with 431k real woodblock glyph instances mined from pool3000 using the same cross-domain-aligned encoder. The 60 evaluation pages are strictly book-disjoint from pool3000, so this enrichment has zero leakage with the test set. With the substitution gate unchanged (thresh = 6.0) and the injected-candidate window widened to 16 for both the baseline and the enriched gallery, the hybrid gallery raises lge_both to 0.7756 (calib) / 0.7763 (rank), a further +0.19 / +0.09 pp over the rendered-font gallery, again concentrated in substitutions, bringing the end-to-end gain over the bare recognizer to +3.66 / +3.73 pp. The retrieval stage is thus lifted by enriching the gallery with real corpus instances without changing the encoder or the decoder; the gallery file (58 MB) is regenerated by build_real_gallery.py and released as supplementary material.

Table 5
Error decomposition, direct configuration, pooled (sub/del/ins per reference char.)

Arm	acc	sub	del	ins	emit	ceil.
base	73.18	19.68	6.07	1.08	95.01	92.86
hog_static	74.70	18.15	6.07	1.08	95.01	92.85
lge_static	74.36	18.50	6.07	1.08	95.01	92.86
lge_image	75.77	17.07	6.07	1.09	95.01	92.84
lge_both	77.37	15.47	6.07	1.09	95.01	92.84

Only substitution moves; deletion and insertion are invariant to the glyph channel, consistent with a lattice that injects one candidate per position and preserves output length.

Because the lattice injects exactly one candidate per position and the decoder emits exactly one character per position, output length is preserved: the predicted length is 23,260 characters for all five arms, identical to the reference minus a constant deletion-insertion difference of 1,221 (|gt|−|hyp|). Deletion and insertion counts themselves vary by at most two characters across arms (a known Levenshtein tie between a single substitution and a delete-insert pair), so only the difference del−ins is an exact identity. The substitution-only movement confirms that the glyph channel affects replacements, not coverage.

6.6. Mechanistic Probe

To localize the gain we align the 4,364 substitution positions (where the recognizer output differs from the ground truth) and rank the true character within each channel's candidate list. Table 6 reports top-K hit rates with a fixed denominator of 4,364 (misses counted as failures, so the figures are comparable across channels); coverage is the fraction of substitution positions for which the channel returns the true character at all.

Table 6
Mechanistic probe: true-character rank at 4,364 substitution positions

Channel	top-1	top-4	top-8	coverage
HOG [12]	10.45	18.68	22.30	27.1%
LGE-static	0.00	17.60	26.26	96.0%
LGE-image	19.41	32.84	39.37	100%

Only top-K hit rates are reported; mean rank is conditional on availability and is not comparable across channels (HOG's low coverage would bias it).

Two findings stand out. First, LGE-static has top-1 of exactly 0.00% by construction: its query is the recognized character's own rendering, whose nearest gallery entry is the character itself, which is by definition not the ground truth at a substitution position. The static channel can therefore never rank the truth first; it can only offer it among ranks 2-8, where the language model chooses. Second, LGE-image places the truth first in 19.41% of positions, the direct source of the +1.18 to +1.49 pp gain. HOG, by contrast, covers only 27.1% of positions (its table is hard-truncated at 20 neighbors with a similarity floor), so its bottleneck is coverage, not ranking, whereas the learned embedding covers 100%.

6.7. Robustness to the Operating Point

The main ablation fixes two operating-point parameters: the substitution gate threshold thresh (default 6.0 nat) and the number of injected glyph candidates per position img_n (default 8). To confirm that the core Δ = lge_image − lge_static is not an artifact of this particular setting, we re-run the direct configuration over a 2x2 grid of (thresh, img_n) $\in \{5.0, 7.0\} \times \{4, 12\}$.

Table 7
Robustness of the core Δ to the operating point (direct configuration; thresh = substitution-gate nat, img_n = injected glyph candidates per position).

thresh / img_n	base	hog	lge_st	lge_img	lge_both	Δ
5.0 / 4	72.18	74.73	72.99	74.51	76.96	+1.52
5.0 / 12	72.18	74.73	73.56	75.22	77.59	+1.66
7.0 / 4	73.71	74.67	74.17	75.20	76.44	+1.03
7.0 / 12	73.71	74.67	74.58	75.69	76.94	+1.10

Direct configuration (a=0.20, b=0.50). All other parameters identical to Table 4; the default operating point (thresh=6.0, img_n=8) is reported there.

Across all four off-default points the core Δ stays positive and lies in the narrow band [+1.03, +1.66] pp (pooled), and the end-to-end lge_both accuracy never drops below 7644.00. The image-over-static gain is therefore robust to the operating point rather than tuned to it.

6.8. Uncertainty-Aware Verification Routing

Although LGE-FIng reduces the substitution error, a residual error rate of 17.64% remains after lge_both on the 60-page set (substitution 3,836, insertion 266, deletion 1,487 positions). These residual errors are not equally addressable. A deletion is a recognition miss: no glyph crop exists at that reference position, so the glyph channel is structurally blind to it. Substitutions and insertions, by contrast, occur at positions that do have a real image block, which the channel can inspect. We therefore partition the residual into capturable (substitution + insertion = 4,102, 73.4%) and uncapturable (deletion = 1,487, 26.6%) with respect to a glyph-channel error detector.

Table 8
Residual error composition after lge_both (60-page set)

Error class	Count	Share	Flagged by glyph router?
Substitution	3,836	68.6%	Yes (crop present)
Insertion	266	4.8%	Yes (crop present)
Deletion	1,487	26.6%	No (no crop)
Capturable (sub+ins)	4,102	73.4%	Yes
Uncapturable (del)	1,487	26.6%	No

Share is over the 5,589 residual errors. The 26.6% deletion share is structurally invisible to any image-conditioned glyph method.

For the capturable majority we train a verification router that, at inference time and without ground truth, flags the most suspicious positions for review (human-in-the-loop) or for a corrective re-decision. The router fuses five uncertainty signals available at inference: the image-channel top-1 cosine g_{top1}, its margin to the second neighbor g_{margin}, the OCR per-position confidence ocr_{conf}, a binary agreement g_{agree} between the image top-1 character and the recognizer output, and a language-model margin lm_{margin}, the log-probability difference between the recognizer output and the image top-1 candidate under a character bigram model trained on in-domain ground truth (the LM scores context only, never the reference). A logistic regressor on z-normalized features yields an out-of-fold AUC of 0.864 on the residual set; the four-signal variant without the LM term reaches 0.765 (and 0.776 when transferred to the OCR layer), so the LM margin is the single largest routing gain.

Table 9
Verification router: precision/recall at review budgets (residual set)

Method	AUC	P@20%	R@50%
Glyph-cos baseline	—	0.314	0.746
OCR-conf baseline	—	0.313	0.707
Glyph-margin baseline	—	0.272	0.720
Heuristic fusion	—	0.379	0.809
Logistic regression (4 signals)	0.765	0.382	0.843
Logistic regression + LM margin (Ours)	0.864	0.616	0.903
+ image–output agreement	0.877	0.602	0.924
+ character-frequency prior (8 signals)	0.890	0.616	0.922

P@K = precision among the K% most-suspicious positions; R@K = recall of all errors captured by reviewing the K% most-suspicious. The proposed router exceeds every single-signal baseline in both metrics. The last two rows add the image–output agreement pair and the frequency prior described below.

At a 20% review budget the router flags positions of which 61.6% are true errors (capturing 69.8% of all errors); reviewing 50% captures 90.3% of errors at 31.9% precision. This is consistent with prior findings that OCR confidence alone is poorly calibrated and improves with multi-signal fusion [24]; character-level uncertainty can be read from the decoder's top-1/top-2 margin [25]; and a language model helps localize errors [26].

Two further inference-time signals sharpen the ranking. The image–output cosine compares the real crop embedding with the gallery entry of the recognizer output itself: a low value says the observed glyph does not look like what the recognizer wrote, and its difference to the image top-1 cosine measures how strongly the image prefers

an alternative. A log character-frequency prior encodes the corpus-level observation that rare characters are harder to recognize. The two families are complementary in an instructive way: the agreement pair raises the review AUC from 0.864 to 0.877 but, used alone as the swap gate, degrades Table 10 from 19.72% to 20.03% CER at the same 10% budget, because the positions it flags are exactly those where the image already disagrees with the output — the same positions where the swap rule then acts and can be wrong. The detector and the swap rule share a failure mode. Adding the frequency prior decorrelates the two and yields the best detector on both criteria (AUC 0.890, R@50% 0.922, gated swap 19.66%).

A per-swap decomposition confirms the diagnosis and suggests the fix. Classifying every applied swap by whether the output was already correct shows that the agreement-pair detector raises the number of swaps that damage a correct output from 170 to 245 while barely changing the number it repairs (191 to 166). Requiring, in addition, the consent of the language model — the swap is applied only where the bigram model also prefers the image candidate over the output — filters out 85% of the damaging swaps (185 to 27) at a cost of eight repaired ones, and the gated swap reaches 19.63% CER, half of the oracle gain. Under the consent gate the choice of detector hardly matters any more (19.57% for the five-signal router), which confirms that the ceiling of the swap is set by the swap rule, not by the detector.

The router does more than rank positions for human review: it is also the safety gate that lets the glyph channel re-decide flagged positions automatically. We add a complementary stroke-topology descriptor (56-D: connected components, Euler number, grid densities, projections) computed on the same crops and rendered gallery, and re-decide a flagged position only when the fused retrieval top-1 differs from the current output and is agreed by both the learned-embedding and the stroke-topology channels (channel-consistent reranking). Table 10 isolates the effect: applying the swap to every position is destructive, because the channels are frequently wrong on positions the recognizer had already answered correctly, whereas gating the same swap by the router yields a net gain at a 10% review budget. The oracle bound applies the swap only where the output is already known to be a substitution error.

Table 10
Error-flag-gated glyph reranking (string-level CER, 60-page set)

Swap strategy	CER (%)	Δ vs baseline (pp)
Baseline (lge_both, no reranking)	20.27	—
Ungated: swap at every position	33.05	+12.79
Router-gated, 10% budget (5 signals)	19.72	−0.55
Router-gated, 10% budget (8 signals)	19.66	−0.61
8 signals + language-model consent gate	19.63	−0.64
Oracle: residual substitutions only	18.98	−1.29

Unified string-level alignment on the 60-page residual set (baseline 20.27% under this accounting). The router-gated rows use the five- and eight-signal routers at a 10% flag budget; the consent-gated row additionally requires the bigram model to prefer the image candidate, and captures 50% of the oracle gain. Budgets above 20% begin to net-harm, which is why the router, not the glyph channel, must decide where the swap is admissible.

The remaining 26.6% deletion errors have no crop and stay invisible to any image-conditioned method, but they are not unreachable. A lightweight detection-level recheck, a second detection pass on the same page with a relaxed box threshold (0.5 to 0.3) and larger unclip ratio, merged into the recognizer output by splicing only the recheck-only segments, recovers 22% of the deletions (196 of 884 aligned deletion characters; 40% when applied to every page without gating) and lowers recognizer-level CER by 0.30 pp net of the extra substitution risk introduced by the spliced characters. The recheck doubles recognition time only on flagged pages and composes with, rather than replaces, the glyph channel; a full treatment of detection-level recovery is beyond the scope of this paper.

6.9. Anatomy of the Residual Substitutions

To locate the remaining headroom we dissect the residual substitutions themselves. Of the 3,836 substitution errors, 3,811 have a distinct output–truth pair and 3,545 are evaluable on the gallery side, i.e., both characters are covered by the embedding vocabulary. For each evaluable pair we ask three questions: does the image query already rank the truth above the output character (recoverable by weight calibration alone); is the truth a near neighbor of the output character in the embedding space (a visual confusion pair); and does the bigram language model prefer the truth given the surrounding context. Table 11 reports the outcome.

Table 11
Where the residual substitutions live (3,545 evaluable pairs)

Recoverability channel	Count	Share
Image already ranks the truth above the output	1,954	55.1%
Truth is the output's nearest gallery neighbor	168	4.7%
Truth within the output's top-5 neighbors	551	15.5%

Truth within the top-20 neighbors	944	26.6%
Language model top-1 recovers the truth	402	11.3%
Reached by none of the three channels	1,105	31.2%

Channels overlap except the last row, which is mutually exclusive with the image-preference test. The median neighbor rank of the truth is 178 (75th percentile 1,720), so the visually confusable tail is broad rather than concentrated.

The dominant output→truth pairs are glyph variants and simplified–traditional counterparts (來→来, 教→教, 淨→淨), which woodblock printing makes visually nearly identical. Three observations follow. First, in 55.1% of the pairs the image channel already prefers the truth, so more than half of the residual substitutions are recoverable by recalibration and gating alone; realizing this preference without collateral damage is precisely what the gating analysis of Table 10 shows to be hard, which is why the oracle swap reaches −1.29 pp rather than the full 6 pp this share would suggest. Second, a quarter of the pairs are visually near misses that a stronger encoder or a confusion-aware loss should reach. Third, 31.2% are reached by neither the image, nor the embedding neighborhood, nor the language model; this hard core, together with the 26.6% deletions, defines where future capacity must go: detection-level recovery and context models longer than a bigram.

7. Discussion and Conclusion

LGE-FIng shows that the query source, not the embedding alone, determines whether a glyph prior helps post-OCR correction. A learned embedding that retrieves against the recognized character's rendering (the static channel) gains only +1.18 pp over the base corrector, because its own rendering is always the top-ranked candidate and the truth is pushed to lower ranks. Querying the real image block instead lifts the truth to rank one in one position out of five, yielding +2.59 pp and, fused with a layout channel, +3.73 pp over the bare recognizer with the real-corpus hybrid gallery. The gain is robust: it holds under four weight calibrations and is largest under a scale-free rank calibration that neutralizes any cosine-scale difference between channels.

The method is cheap and portable by design. The encoder is a 2M-parameter convolutional network trained for about six epochs per seed on a single GPU; the gallery is built once from font renderings and requires no additional transcription. At inference the channel adds a nearest-neighbor lookup per character and a fixed number of lattice candidates, and it degrades exactly to the prior corrector when disabled. This makes LGE-FIng a drop-in component for any recognizer that exposes per-character glyph images, which is the typical industrial digitization pipeline.

Two limitations bound the claim. First, the +3.73 pp end-to-end gain (real-corpus hybrid gallery) is measured against a PP-OCRv5 backbone whose own swap from PP-OCRv4 contributed +12.02 pp [15]; recognition improvement remains the larger lever, and LGE-FIng is complementary to it. Second, the gallery covers 88.5% of the dictionary because some characters cannot be rendered by the available fonts; this affects only rare characters outside the corrector's working vocabulary and is not the current bottleneck. The anatomy of Section 6.9 sharpens this picture: 55.1% of the residual substitutions are already ranked first by the image channel and await calibration and gating, a quarter are visually near misses for a stronger encoder, and the 31.2% hard core plus the deletions define the next frontier. Future work includes learned re-ranking of the gallery neighbors with image features and extending the cross-domain alignment to dialect and seal-script variants.

LGE-FIng makes the query source the variable that decides whether a glyph prior helps: retrieving against the recognized character's own rendering strands the truth at lower ranks, whereas retrieving against the real glyph image puts it first often enough to lift accuracy by +3.73 pp end-to-end. The gain is isolated, calibrated, and stable across every weight mapping we tried, and it composes with a real-corpus gallery at zero test leakage. The 26.6% of errors that remain are deletions that no image-conditioned method can see and are left to detection-level or language-model gap-filling.

CRediT authorship contribution statement

Laixin Shen: Conceptualization, Methodology, Software, Investigation, Data curation, Writing – original draft. Fan Yang: Validation, Visualization, Writing – review & editing, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY

The page-alignment files and the 60-page evaluation bundle are available from the corresponding author on reasonable request. The source page images are subject to an institutional subscription license and cannot be redistributed.

ACKNOWLEDGMENT

This work was supported by the Anhui Provincial University Natural Science Research Project (Grant No. 2025AHGXZK30889).

DECLARATION OF GENERATIVE AI AND AI-ASSISTED TECHNOLOGIES IN THE MANUSCRIPT PREPARATION PROCESS

During the preparation of this work the authors used WorkBuddy (an AI agent powered by the GLM large language model) in order to assist with experiment code development, manuscript drafting, and language editing. The authors reviewed and edited the output as needed and take full responsibility for the content of the published article.

REFERENCES

[1] Z. Zhou, Y. Sun, H. He, et al., “TongGuOCR: A layout-aware and token-augmented OCR MLLM for Chinese historical documents,” arXiv:2608.07917, 2026.

[2] H. Yu, Y. Wu, F. Shi, et al., “Benchmarking vision-language models on Chinese ancient documents: From OCR to knowledge reasoning,” arXiv:2509.09731, 2025.

[3] R. Smith, “An overview of the Tesseract OCR engine,” in Proc. 9th Int. Conf. Document Analysis and Recognition (ICDAR), Curitiba, Brazil, 2007, pp. 629–633, doi: 10.1109/ICDAR.2007.4376991.

[4] Y. Du, C. Li, R. Guo, et al., “PP-OCR: A practical ultra lightweight OCR system,” arXiv:2009.09941, 2020.

[5] M. Li, T. Lv, J. Chen, et al., “TrOCR: Transformer-based optical character recognition with pre-trained models,” in Proc. AAAI Conf. Artificial Intelligence, vol. 37, no. 11, 2023, pp. 13094–13102, doi: 10.1609/aaai.v37i11.26538.

[6] Y. Zheng, Y. Chen, X. Wang, D. Qi, and Y. Yan, “Ancient Chinese character recognition with improved Swin-Transformer and flexible data enhancement strategies,” Sensors, vol. 24, no. 7, art. no. 2182, 2024, doi: 10.3390/s24072182.

[7] A. Lee, H. Yu, and G. Min, “An algorithm of line segmentation and reading order sorting based on adjacent character detection: A post-processing of OCR for digitization of Chinese historical texts,” J. Cultural Heritage, vol. 67, pp. 80–91, 2024, doi: 10.1016/j.culher.2024.02.001.

[8] G. Chiron, A. Doucet, M. Coustaty, and J.-P. Moreux, “ICDAR2017 competition on post-OCR text correction,” in Proc. 14th IAPR Int. Conf. Document Analysis and Recognition (ICDAR), vol. 1, 2017, pp. 1423–1428, doi: 10.1109/ICDAR.2017.232.

[9] M. Ehrmann, E. Boros, J. Opitz, et al., “ICDAR 2026 HIPE-OCRepair competition on LLM-assisted OCR post-correction for historical documents,” arXiv:2607.08143, 2026.

[10] I. H. Witten and T. C. Bell, “The zero-frequency problem: Estimating the probabilities of novel events in adaptive text compression,” IEEE Trans. Information Theory, vol. 37, no. 4, pp. 1085–1094, 1991, doi: 10.1109/18.87000.

[11] N. Nguyen, T. Nguyen, V. Tran, M.-T. Tran, T. D. Ngo, T. H. Nguyen, and M. Hoai, “Dictionary-guided scene text recognition,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2021, pp. 7379–7388, doi: 10.1109/CVPR46437.2021.00730.

[12] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition (CVPR), vol. 1, 2005, pp. 886–893, doi: 10.1109/CVPR.2005.177.

[13] D. Wang, C. Liu, Z. Zhu, J. Liu, H. Hu, S. Shen, and B. Li, “SikuBERT and SikuRoBERTa: Construction and application of pre-training models of ‘Siku Quanshu’ oriented to digital humanities,” Library Tribune, vol. 42, no. 6, pp. 31–43, 2022, doi: 10.3969/j.issn.1002-1167.2022.06.005. (in Chinese)

[14] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, “ArcFace: Additive angular margin loss for deep face recognition,” in Proc. IEEE/CVF Conf. Computer Vision and Pattern Recognition (CVPR), 2019, pp. 4690–4699, doi: 10.1109/CVPR.2019.00482.

[15] C. Cui, T. Sun, M. Lin, et al., “PaddleOCR 3.0 technical report,” arXiv:2507.05595, 2025.

[16] H.-Y. Ma, H.-H. Huang, and C.-L. Liu, “Reading between the lines: Image-based order detection in OCR for Chinese historical documents,” in Proc. AAAI Conf. Artificial Intelligence, vol. 38, no. 21, 2024, pp. 23808–23810, doi: 10.1609/aaai.v38i21.30572.

[17] J. Bourne, “CLOCR-C: Context leveraging OCR correction with pre-trained language models,” arXiv:2408.17428, 2024.

[18] E. Boroş, M. Ehrmann, M. Romanello, S. Najem-Meyer, and F. Kaplan, “Post-correction of historical text transcripts with large language models: An exploratory study,” in Proc. 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL), 2024, pp. 133–159, doi: 10.18653/v1/2024.latechclfl-1.14.

[19] Z.-C. Lin and W.-H. Liao, “Decoding-time fusion of OCR and large language models for traditional Chinese historical document recognition,” in Proc. 28th Int. Conf. Pattern Recognition (ICPR), Lyon, France, 2026, Lecture Notes in Computer Science, vol. 16819, pp. 48–62, doi: 10.1007/978-3-032-31404-8_4.

[20] C.-J. Hsu, Y.-C. Chen, F.-T. Liao, et al., “Let’s fuse step by step: A generative fusion decoding algorithm with LLMs for robust and instruction-aware ASR and OCR,” in Findings of the Association for Computational Linguistics: ACL 2025, 2025, pp. 24959–24973, doi: 10.18653/v1/2025.findings-acl.1281.

[21] J. Kanerva, C. Ledins, S. Käpyaho, and F. Ginter, “OCR error post-correction with LLMs in historical documents: No free lunches,” arXiv:2502.01205, 2025.

[22] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, “A discriminative feature learning approach for deep face recognition,” in Proc. European Conf. Computer Vision (ECCV), 2016, pp. 499–515, doi: 10.1007/978-3-319-46478-7_31.

[23] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in Proc. Int. Conf. Machine Learning (ICML), 2015, pp. 97–105.

[24] M. Cuper, C. van Dongen, and T. Koster, "Unraveling confidence: Examining confidence scores as proxy for OCR quality," in Proc. Int. Conf. Document Analysis and Recognition (ICDAR), 2023, Lecture Notes in Computer Science, vol. 14191, pp. 104–120, doi: 10.1007/978-3-031-41734-4_7.

[25] G. H. Kim, J. Bak, and H. Choi, "Uncertainty-aware processing for OCR using dictionary routing and candidate selection," Electronics Letters, vol. 62, no. 1, 2026, art. no. e70630, doi: 10.1049/ell2.70630.

[26] A. Hemmer, M. Coustaty, N. Bartolo, and J.-M. Ogier, "Confidence-aware document OCR error detection," arXiv:2409.04117, 2024.

[27] BYVoid, "OpenCC: Open Chinese Convert," open-source project. [Online]. Available: https://github.com/BYVoid/OpenCC

[28] Dingxiu Ancient Books Full-Text Retrieval Platform, Beijing Hanhai Boya Technology Co., Ltd. [Online]. Available: http://www.ding-xiu.com

BIOGRAPHIES

Laixin Shen is a professor at the Huizhou Culture Research Center of Anhui Province and the School of Information Engineering, Huangshan University, Anhui, China. His research interests include cultural heritage digitization, document image analysis and optical character recognition for historical Chinese texts. E-mail: slx965@163.com.

Fan Yang received the M.S. degree from Hefei University of Technology, Hefei, China, in 2007. She is currently a lecturer with the School of Information Engineering, Huangshan University, Anhui, China. Her research interests include data mining, big data, pattern recognition and machine learning. E-mail: fanyang80@163.com.