symmetry

MDPI

Article

Chinese and English Scene Text Image Super-Resolution with Constrained LLM Correction of OCR Priors and Mamba-Assisted Diffusion

Xinyu Qiu [1], Jingchao Liu [1,2,*], Pengli Xu [1], Chenghao Lu [1] and Yu Fu [1]

[1] School of Computer Science, Xijing University, Xi'an 710123, China; 2408540402033@stu.xijing.edu.cn or xyqiu0211@gmail.com (X.Q.); 2608540402002@stu.xijing.edu.cn (P.X.); 2608540402001@stu.xijing.edu.cn (C.L.); 20180141@xijing.edu.cn (Y.F.)

[2] School of Computer Science and Technology, Beijing Jiaotong University, Beijing 100044, China

* Correspondence: liujingchao@xijing.edu.cn

Abstract

Scene text image super-resolution must preserve character identity and glyph geometry, including approximate symmetry where present and asymmetric details that distinguish characters. Erroneous optical character recognition (OCR) priors can distort characters during generative reconstruction. We combine constrained large language model (LLM) correction and Mamba-based visual sequence modeling in a conditional residual diffusion method. The LLM proposes replacements within OCR candidate and edit constraints; deterministic validation rejects invalid suggestions and retains the original transcription. Corrected text, position, and low-resolution structural priors are fused by sample-level reliability gating and injected through cross-attention. A bottleneck Mamba block models horizontal visual dependencies. Under separately trained TextZoom and Real-CE protocols, the method achieves peak signal-to-noise ratios of 25.06 and 24.38 dB and exact-match accuracies of 83.4% and 50.2%, respectively. Five-seed paired ablations show that constrained OCR correction improves full-model reconstruction by 0.07 dB in PSNR and 0.6 percentage points in exact-match accuracy on both TextZoom and Real-CE, with the corresponding 95% confidence intervals excluding zero. Compared with the tested lightweight Transformer, Mamba reduces reconstruction latency by 7.82 ms with comparable quality. Independent recognition evaluation supports improved readability. The results support constrained correction and complementary visual conditioning.

Keywords: scene text image super-resolution; OCR correction; residual diffusion; Mamba; glyph symmetry and asymmetry; prior reliability

Received: 27 September 2026
Revised: 21 October 2026
Accepted: 03 November 2026
Published: 5 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

1.1. Background and Motivation

Scene text image super-resolution (STISR) reconstructs a readable high-resolution image from a low-resolution observation. Real scenes combine distance-related loss of detail, motion or defocus blur, noise, occlusion, and uneven illumination. STISR must preserve character identity under these degradations: a missing stroke, an unintended connection, or a small displacement can change the transcription even when the image appears sharper. TextZoom and Real-CE provide paired data for evaluating image quality

and recognition performance on real English text lines and Chinese–English scenes, respectively [1,2].

Character geometry involves both approximate symmetry and identity-defining asymmetry. In particular typefaces, certain glyphs or local stroke arrangements exhibit approximately reflected or balanced structures, whereas other characters are distinguished by unequal component positions, opening directions, or asymmetric stroke configurations. Image degradation can disrupt these relationships in different ways: blur and missing strokes may destroy an originally balanced arrangement, while generative reconstruction may incorrectly regularize an asymmetric detail into a more visually plausible but character-inaccurate pattern.

Consequently, faithful scene text super-resolution should preserve approximate symmetry where supported by the observed character and retain distinctive asymmetry where it determines character identity. The objective is not to make every reconstructed glyph symmetric, but to recover the structural relationships supported by the input image. This geometric perspective motivates the complementary use of textual identity, OCR-derived position, and LR-derived structural evidence in the proposed reconstruction framework.

Text priors provide content information that may be difficult to recover from degraded pixels alone. Text attention, recognition-guided reconstruction, and dual-prior modulation have improved how reconstruction uses this information [3–5]. Text-conditioned diffusion and prior-enhanced reconstruction extend such guidance to generative restoration [6,7]. Because the conditioning text often comes from the same degraded input, reconstruction can reinforce an OCR error and produce a plausible but incorrect character. The prior's reliability must therefore be considered during restoration.

Large language models (LLMs) can suggest corrections that fit the linguistic context, but these suggestions may not match the image. Short strings, names, numbers, and mixed-language text are particularly difficult to judge from context alone. Exploratory research on OCR post-correction reports that LLM performance varies across settings [8]. A correction procedure must preserve the available OCR evidence and permit rejection of suggestions that fall outside it.

Visual context remains necessary after text correction. Convolution captures local strokes, and sequence models propagate information across character regions. Mamba provides a selective state-space formulation for sequence modeling [9]; MambaIR adapts this approach to image restoration [10]. The effect of such a module depends on its scan direction, sequence length, and placement within the network. We examine a bottleneck configuration that complements convolution and cross-attention, comparing it with a lightweight Transformer at the same location.

These considerations define the core research problem of this study: how to exploit uncertain OCR-derived text priors for scene text super-resolution while limiting the propagation of incorrect character identities into the reconstruction. This problem involves two distinct decisions. Before reconstruction, a correction proposal must be checked against the available OCR candidates rather than accepted solely for linguistic plausibility. During reconstruction, even an accepted or unchanged transcription may remain incorrect, so its contribution must be adjusted in conjunction with positional and LR structural evidence. The objective is therefore reliability-aware use of text priors, rather than unrestricted language-model rewriting or the assumption that a visually sharper image necessarily preserves the correct characters.

1.2. Proposed Idea

Our previous publication concerned OCR priors and cross-scale attention [11]. The present method organizes text-prior preparation and visual reconstruction around the

reliability of the conditioning information. Before reconstruction, the OCR–LLM interface retains the original transcription, alternative character candidates, and confidence information. The LLM provides linguistic suggestions within this candidate space, while a separate deterministic validator enforces the editing constraints and rejects invalid proposals. This separation allows linguistic context to be considered without treating a plausible correction as automatically trustworthy. Cached proposals, decisions, and adopted strings make the correction stage inspectable independently of the reconstructed images.

During reconstruction, the adopted text is combined with positional and LR structural priors because character identity, spatial layout, and observed stroke evidence provide different information. The textual branch supplies character-content cues; the positional branch associates these cues with OCR locations; and the structural branch supplies image-derived evidence that does not depend on the linguistic plausibility of the transcription. Sample-level reliability gating adjusts the relative contributions of these branches using OCR confidence, correction status, and visual evidence. Cross-attention then aligns the fused conditions with visual features. Gating and cross-attention therefore serve different purposes: the former adjusts the contribution of each prior branch, whereas the latter performs condition-to-feature alignment.

RRDB provides an initial visual reconstruction, and the conditional diffusion model estimates the residual relative to that image. A row-wise Mamba block at the conditioned U-Net bottleneck supplements horizontal visual sequence modeling; it neither corrects OCR text nor replaces the text encoder. These are established building blocks with distinct roles in the proposed task-specific design. The methodological contribution lies in linking constrained, traceable text adoption to reliability-weighted visual conditioning within residual reconstruction, rather than in introducing a new general-purpose LLM, diffusion formulation, or state-space operator. The connection to symmetry and asymmetry lies in the geometry-sensitive reconstruction task: LR structural features describe observed stroke magnitude, orientation, and spatial arrangement, which can contain approximately symmetric patterns as well as character-defining asymmetries. The proposed framework incorporates this evidence alongside textual and positional conditions without imposing reflection symmetry on every glyph. It does not introduce a symmetry-specific loss or claim reflection equivariance; instead, its evaluated contribution concerns reconstruction quality and recognition reliability when uncertain textual priors are combined with visual structural evidence.

1.3. Contribution

The main contributions of this paper are summarized as follows:

1. OCR-supported and auditable prior correction. We formulate a constrained OCR–LLM interface that retains candidate uncertainty, permits bounded substitutions, and separates linguistic proposal generation from deterministic acceptance, rejection, and fallback. Cached proposals and decisions make the adopted text traceable and allow correction quality to be evaluated separately from its downstream reconstruction effect.
2. Complementary conditioning for residual visual reconstruction. We combine adopted text, OCR-derived positions, and LR-derived structural evidence to represent character content, spatial layout, and observed strokes. Cross-attention aligns these conditions with visual features in an RRDB-initialized residual diffusion framework, while a bottleneck row-wise Mamba block supplements horizontal visual context. The contribution is their task-specific organization, rather than a new generic diffusion or sequence-modeling operator.
3. Sample-level reliability adjustment after text adoption. Because satisfying the correction constraints does not establish that a transcription is correct, we use OCR

confidence, correction status, and visual evidence to adjust the relative contributions of the textual, positional, and structural branches. Comparisons with equal and fixed global weights examine this reliability-aware conditioning mechanism within the tested reconstruction configuration.

The evaluation uses separately trained TextZoom and Real-CE models, image-quality and recognition metrics, component ablations, an independent recognizer, and resource measurements. These analyses assess the proposed integration and its observed limits.

2. Related Work

2.1. Text Priors and Diffusion-Based Reconstruction

Scene text image super-resolution requires cues about both visual appearance and character content. Text-oriented methods supplement image features with recognition or structural information. TATT uses text attention to handle spatial deformation, TPGSR introduces text priors into reconstruction, and DPMN uses dual-prior modulation to improve character recovery [3–5]. Our earlier publication addressed OCR priors and cross-scale attention [11]. The present study examines the condition-construction stage explicitly: whether an OCR candidate should be corrected, which substitutions are allowed, and how accepted or rejected suggestions are passed to the reconstruction network. It then evaluates the downstream effect of replacing raw OCR text with constrained-correction text under the same base conditioning configuration.

Diffusion models iteratively refine text images under conditional guidance. Text-conditioned diffusion has been applied to scene text restoration [6], and PEAN combines diffusion-generated priors with attention-based reconstruction [7]. Recent studies also address errors in the conditioning information. TextSR examines multilingual OCR-guided diffusion super-resolution and the effects of inaccurate OCR conditions [12]. The PRISM preprint studies prior rectification during training alongside uncertainty-aware structural modeling [13]. Our work addresses this established problem of prior reliability through an explicit correction interface. The interface retains OCR candidates, limits permitted edits, and records whether an LLM suggestion is accepted or rejected, allowing text correction to be evaluated separately from image reconstruction.

2.2. LLM-Assisted OCR Post-Correction

OCR post-correction revises recognized strings using character-level evidence and linguistic context. LLMs can propose plausible alternatives, although their effectiveness depends on the text domain and prompting strategy [8]. A plausible string may still disagree with the source image. A valid rare word, proper name, or mixed-language string may have a lower language-model probability than a familiar but incorrect alternative. In super-resolution, a mistaken string can affect both the transcription and the strokes generated during reconstruction.

Our correction procedure uses the evidence available in the low-resolution OCR output. The LLM receives candidate characters, confidence information, permitted edit positions, and language information. It neither inspects the input image nor produces a new visual transcription. Its response is a candidate proposal rather than calibrated confidence: the procedure does not treat the LLM's token probabilities as correction reliability. A separate validator checks character membership, output length, edit distance, and candidate coverage before text encoding. A failed check triggers fallback to the original OCR output. The cached decision makes adoption inspectable, but the candidate set and equal-length editing rule limit what can be corrected. Missing or extra characters, and true characters absent from the candidate set, remain unresolved by this interface.

2.3. State Space Models for Image Restoration

Mamba propagates information along a sequence through selective state space updates with input-dependent parameters [9]. MambaIR combines state space modeling with local feature enhancement for image restoration [10]. Vision Mamba and VMamba explore bidirectional and multidirectional visual processing, respectively, illustrating the role of scan design when mapping two-dimensional features to sequences [14,15]. We use one left-to-right scan per feature row at the diffusion U-Net bottleneck. This block operates on visual states after conditional alignment; it does not replace the text encoder or determine whether an OCR correction is accepted. Convolution and skip connections handle local features, while Mamba propagates horizontal context. Its assessment is limited to the tested bottleneck configuration and the accompanying quality and resource measurements.

3. Proposed Method

3.1. Problem Formulation and Overview

Given a low-resolution image I_{LR}, we aim to reconstruct a high-resolution image I_{SR} that preserves the text content and observed glyph structure. Training uses paired high-resolution targets I_{HR} and available transcriptions. At inference, textual, positional, and structural evidence is obtained from I_{LR} and its cached OCR outputs. The OCR–LLM correction stage additionally uses the language-category metadata specified by the dataset protocol, as described in Section 3.2. A residual-in-residual dense block (RRDB) reconstruction network [16] first produces I_{init}, and the diffusion model estimates the residual relative to this reconstruction:

$$I_{\mathrm{init}} = G_{\mathrm{RRDB}}(I_{\mathrm{LR}}; s), I_{\mathrm{SR}} = I_{\mathrm{init}} + \hat{x}_0 \quad (1)$$

Here, G_{RRDB} is the initial reconstruction network, s is the upscaling factor, and $\hat{x}_0$ is the sampled residual. The clean training residual is $x_0 = I_{\mathrm{HR}} - I_{\mathrm{init}}$. We train separate models for the ×2 TextZoom and ×4 Real-CE protocols. The condition pathway combines OCR evidence extraction and constrained LLM correction with three prior encoders and reliability gating. The visual pathway combines the initial RRDB reconstruction with a diffusion U-Net. Cross-attention injects conditions at the bottleneck (CA-B) and at one-quarter resolution in the decoder (CA-M). These sites share prior sources and have independent projections. A single Mamba block processes the conditioned bottleneck features. Figure 1 summarizes the condition and reconstruction pathways.

3.2. Constrained LLM-Based OCR Correction

The correction stage retains the uncertainty in OCR outputs. A frozen PaddleOCR detector and recognizer extract the original string y, character candidate probabilities, confidence scores, and text locations from I_{LR}. For complete scenes, detection precedes recognition, and each transcription retains its text-instance index and corresponding box. The OCR configuration and character-normalization rules are fixed across comparisons. At each valid decoded position i, we retain a Top-K candidate set C_i; positions with high confidence and no candidate conflict are protected from editing. With connectionist temporal classification (CTC) decoding, frame-level evidence is aggregated to output character positions under the recognizer's alignment rule and CTC mapping of blanks and repeated labels. Blank symbols and duplicate paths are not passed to the LLM as visible characters.

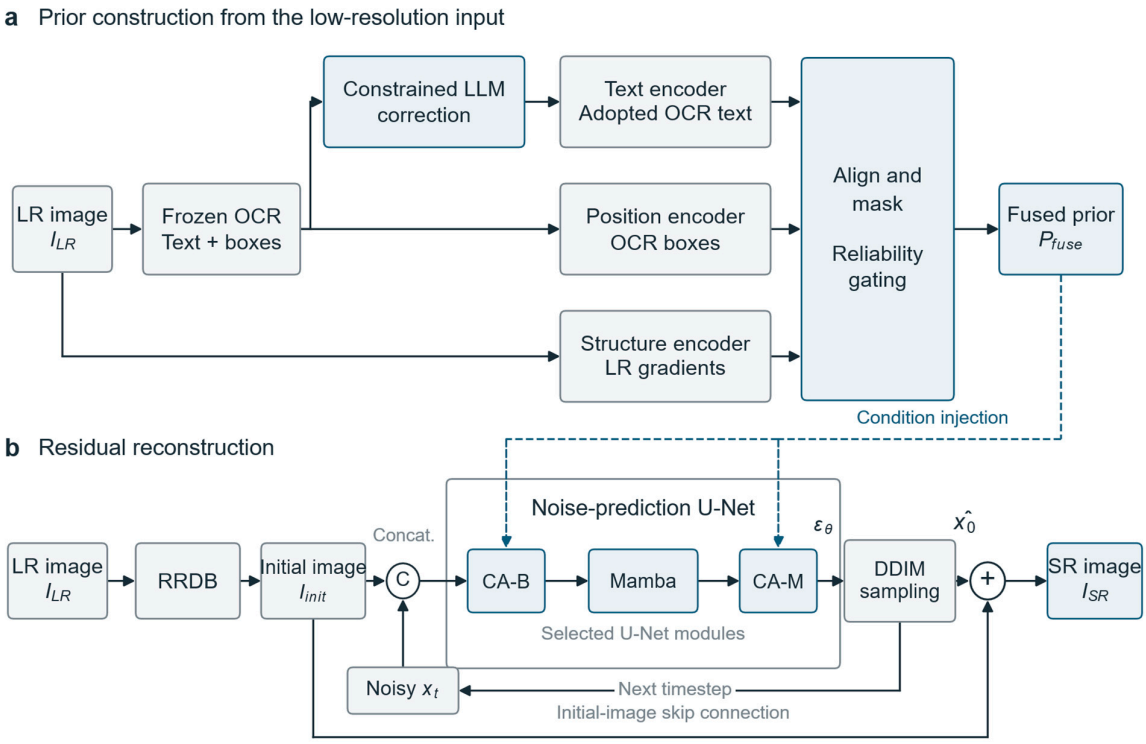


Figure 1. Overview of the reconstruction framework. (**a**) Optical character recognition (OCR) provides textual evidence for constrained large language model (LLM) correction and position encoding. The structural prior comes directly from the low-resolution (LR) image. Aligned and gated priors condition two cross-attention sites (dashed lines). (**b**) A residual-in-residual dense block (RRDB) network produces the initial image. The U-Net predicts noise; denoising diffusion implicit model (DDIM) sampling estimates the residual, which is added to the initial image to form the super-resolved (SR) output. CA-B and CA-M denote bottleneck and decoder cross-attention, respectively. Only selected U-Net modules are shown; C denotes channel concatenation.

Character probabilities were calibrated using temperature scaling on an internal calibration subset drawn from the training data of TextZoom and Real-CE. The calibration subset was kept separate from the final evaluation sets, and the OCR recognizer parameters remained frozen throughout calibration. A single temperature parameter was determined by minimizing the negative log-likelihood of the reference characters under the fixed OCR alignment and decoding procedure.

The selected temperature was $T = 1.18$, and this value was held fixed for both evaluation datasets. Specifically, OCR logits were divided by T before applying softmax to obtain calibrated character-candidate probabilities. This transformation adjusts the sharpness of the probability distribution without changing the ranking of candidates. The calibrated probabilities were consistently used for correction triggering, candidate scoring, and reliability estimation. The value $T = 1$ denotes the uncalibrated setting rather than the fitted calibration temperature.

Sample-level OCR confidence was calculated as the geometric mean of the calibrated probabilities assigned to valid decoded characters, thereby reducing the length dependence associated with the direct product of character probabilities:

$$c_{\mathrm{ocr}} = \exp\left(\frac{1}{L}\sum_{i=1}^{L}\log\left(p_{i,y_i} + \varepsilon\right)\right) \quad (2)$$

Here, L is the decoded string length, the probability inside the logarithm corresponds to the i-th decoded character, and ε is a numerical stabilizer. Empty transcriptions are marked invalid. We set $K = 5$ and the sample-confidence threshold $\tau_c = 0.65$. Candidate conflicts are quantified using the gap between the two highest calibrated probabilities at each position:

$$u_i = 1 - \left(p_{i,(1)}^{(T)} - p_{i,(2)}^{(T)}\right) \tag{3}$$

In Equation (3), $p_{i,(1)}^{(T)}$ and $p_{i,(2)}^{(T)}$ denote the highest and second-highest calibrated OCR candidate probabilities at character position i, respectively. Define their probability gap as $\Delta_i = p_{i,(1)}^{(T)} - p_{i,(2)}^{(T)}$, so that the uncertainty in Equation (3) is $u_i = 1 - \Delta_i$. For a valid OCR string, a candidate conflict is detected when $\min_i \Delta_i \leq 0.20$, equivalently when $\max_i u_i \geq 0.80$, with i ranging over valid decoded character positions. Correction is triggered when either the sample-level OCR confidence is below $\tau_c = 0.65$ or a candidate conflict is detected. Samples satisfying neither condition retain the original OCR transcription without invoking the LLM. Only editable positions expose alternative candidates, so the full Cartesian product of candidate sets is not explicitly enumerated. Triggering correction initiates candidate generation but does not guarantee acceptance; a proposed correction is adopted only after passing all deterministic validation rules.

Qwen2.5-7B-Instruct [17] runs offline in bfloat16 (BF16) precision. Its structured prompt contains the OCR string, editable positions, candidate characters and probabilities, character uncertainty, sample confidence, language type, and restricted contextual information. The task field requests limited OCR correction; constraint fields specify the permitted character set, equal-length rule, edit budget, and candidate coverage. The maximum edit budget is set to $D_{\max} = 2$ character substitutions per OCR string. Specifically, no more than two character positions may be changed relative to the original OCR transcription. Insertions and deletions are prohibited, and each accepted candidate must have the same length as the original OCR string. A candidate is rejected if its edit distance exceeds $D_{\max}$, if a replacement falls outside the permitted OCR candidate set, or if a protected position is modified. These deterministic constraints prevent unrestricted text rewriting and restrict the adopted correction to a bounded set of OCR-supported substitutions. The response contains the proposed text, whether an edit is proposed, edited positions, a reason code, and refusal status.

A simplified prompt template is shown below to clarify the input and output interface. The values in braces are filled from the cached OCR results and the dataset-defined language constraints.

Prompt template

"You are an OCR post-correction assistant. Correct the OCR transcription only when the provided visual-recognition evidence and permitted candidates support a change. Do not invent, expand, translate, or freely rewrite the text.

Language type: {language_type}

Original OCR transcription: {ocr_text}

Sample OCR confidence: {sample_confidence}

Editable positions and permitted replacement candidates: {editable_positions_and_candidates}

Character-level uncertainty information: {character_uncertainty}

Maximum edit budget: {D_max}

Constraints:

Preserve the original string length.

Modify only editable positions.

Use only the permitted candidate characters at edited positions.

Do not insert or delete characters.

If no candidate is sufficiently supported, retain the original transcription.

Return a JSON object with the fields: proposed_text, edit_proposed, edited_positions, decision_status, reason_code, and refusal_status. Do not include explanatory prose outside the JSON object."

This template illustrates the constrained interface; the deterministic validator, rather than the prompt alone, enforces the hard constraints before a proposed correction can be adopted.

Explanations are excluded from the adopted string. The language type and permitted character set are supplied through the sample-level language metadata specified by the dataset protocol, rather than inferred by the LLM. This information is included in the structured prompt before correction, and the same language constraints are applied during deterministic validation. Mixed Chinese–English samples are not forced into an exclusively Chinese or English category. Their permitted character set includes the Chinese characters, Latin letters, digits, and punctuation allowed by the protocol. Candidate substitutions must satisfy both the permitted character-set constraints and the OCR candidate constraints at each editable position. The LLM is not allowed to translate the text or freely rewrite a mixed-language transcription.

The correction procedure reads the OCR transcription and associated language metadata, constructs the structured prompt, and generates a proposal only when the correction-triggering criteria are met. The proposal is then checked by the deterministic validator before acceptance. The LLM has no access to LR or HR images, ground-truth transcriptions, test-answer lexicons, or HR-derived OCR. Ground-truth transcriptions and HR image information are not used to determine the language type. All corrections are generated before super-resolution training and stored offline, with no per-test-sample prompt adjustment.

Qwen2.5-7B-Instruct runs in BF16 precision and uses greedy decoding with do_sample = False, num_beams = 1, max_new_tokens = 128, and repetition_penalty = 1.0. Its structured JSON response contains proposed_text, edit_proposed, edited_positions, decision_status, reason_code, and refusal_status. Invalid or incomplete JSON outputs trigger fallback to the original OCR transcription.

Under the deterministic decoding configuration, Qwen2.5-7B-Instruct generates a single structured correction proposal for each triggered OCR transcription. The proposed transcription is extracted from the proposed_text field and subjected to deterministic validation. The validation procedure checks language compatibility, character-set membership, string-length consistency, protected positions, the maximum edit budget, and membership of each substituted character in the corresponding Top-K OCR candidate set. Insertions, deletions, and changes outside editable positions are prohibited.

The original OCR transcription is always retained as an admissible no-edit alternative. When the generated proposal satisfies all validation rules, the original transcription and the validated proposal form a feasible candidate set containing at most two distinct strings. The candidate-selection score combines linguistic evidence, calibrated OCR character probabilities, and the edit-distance penalty. The higher-scoring transcription is adopted, with ties resolved in favor of the original OCR output. Invalid proposals are rejected, and parsing or execution failures trigger rollback to the original transcription.

Thus, greedy decoding is used to generate the structured proposal, while the scoring function is used to compare the feasible transcriptions. The procedure does not enumerate the Cartesian product of OCR character candidates or require stochastic decoding to obtain multiple proposals. Candidate selection is defined as follows:

$$S(z) = \lambda_{\mathrm{lm}} log p_{\mathrm{LLM}}(z \mid q) + \lambda_{\mathrm{ocr}} \sum_{i=1}^{|z|} \log\left(p_{i,z_i}^{(T)} + \varepsilon\right) - \lambda_{\mathrm{ed}} d_{\mathrm{edit}}(z, y), \quad z \in \Omega \tag{4}$$

Here, z denotes a feasible candidate transcription, s_{ocr} is the original OCR transcription, and q is the fixed structured correction prompt. The feasible candidate set consists of the original transcription and, when deterministic validation succeeds, the LLM-generated proposal. The correction-stage scoring coefficients are $\lambda_{\mathrm{lm}} = 1.0$, $\lambda_{\mathrm{ocr}} = 1.0$, and $\lambda_{\mathrm{ed}} = 0.5$, with a maximum edit budget of $D_{\max} = 2$. These coefficients are used exclusively for candidate selection and are independent of the reconstruction-loss coefficients defined in Section 3.6. The linguistic term evaluates the LLM's conditional preference for a candidate transcription, the OCR term incorporates calibrated character-level recognition evidence, and the edit-distance penalty discourages unnecessary substitutions. All scoring parameters remain fixed across the evaluated datasets and language groups.

The LLM sequence log-probability in Equation (4) is calculated from autoregressive token-level conditional probabilities using teacher-forced candidate evaluation. Each feasible transcription is tokenized using the Qwen2.5-7B-Instruct tokenizer and evaluated under the same fixed correction prompt and output-field prefix. The prefix terminates immediately before the value of the proposed_text field, ensuring that candidate transcriptions are evaluated under equivalent linguistic contexts.

For each candidate token, the corresponding model logits are converted to log-probabilities using log-softmax. The conditional log-probabilities assigned to the candidate tokens are then summed to obtain the sequence score. Prompt tokens and fixed JSON-format tokens are excluded from this calculation. Both the original OCR transcription and the validated LLM proposal are evaluated using the same procedure, without generating additional candidate sequences.

The OCR evidence term is calculated from calibrated character probabilities at the corresponding decoded character positions, with numerical stabilization applied before taking logarithms. The resulting LLM sequence log-probability represents a conditional linguistic preference rather than a calibrated probability that the transcription agrees with the source image. Consequently, candidate adoption depends jointly on the linguistic score, OCR evidence, edit penalty, and deterministic constraints.

When correction is not triggered, the original OCR transcription is retained without invoking the LLM. When correction is triggered, the structured proposal is first subjected to deterministic validation. If the proposal passes validation, its score is compared with that of the original OCR transcription using Equation (4). The higher-scoring transcription is adopted, with ties resolved in favor of the original transcription. The decision label accept indicates adoption of a valid corrected proposal, keep indicates retention of the original transcription, reject indicates failure of a deterministic constraint, and rollback indicates a parsing or execution exception. A valid proposal whose score does not exceed that of the original transcription is recorded as keep, rather than as a validation rejection. Figure 2 provides a high-level overview of correction triggering, deterministic validation, and fallback. The detailed candidate-scoring and final-selection rules are defined separately in Equation (4).

The frozen cache stores candidates, original, proposed and adopted strings, edited positions, text-instance identifiers, decision status, and versions of the OCR model, LLM, prompt, and validation rules. Experiments read these fixed records. Miscorrection counts originally correct OCR strings that become incorrect after an accepted edit, using initially correct strings as the denominator. Rejection records a failed constraint check, whereas rollback records a parsing or execution exception; both rates use all evaluated samples as the denominator. These text-level events are distinct from a downstream reconstruction failure, defined as a higher per-sample CER for the full model than for RRDB under the

https://doi.org/10.3390/sym18111860

same evaluation recognizer. OCR and LLM preprocessing is excluded from online reconstruction timing, although it remains part of the full system cost.

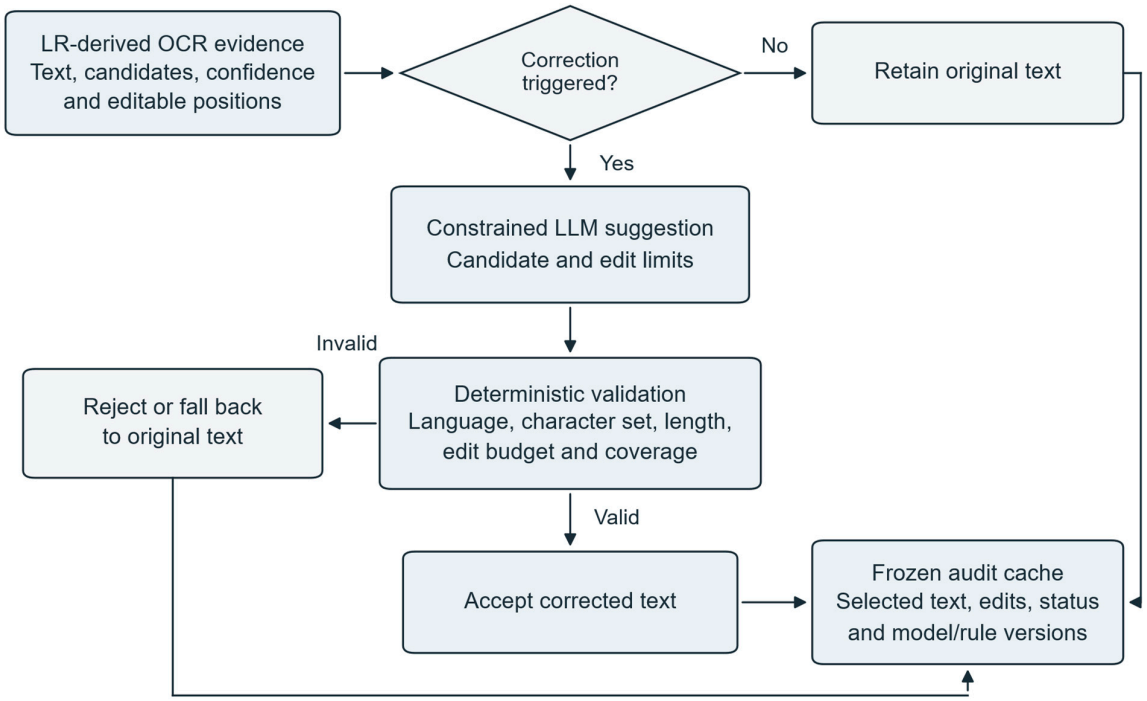


Figure 2. Constrained OCR–LLM correction. OCR candidates specify editable positions and permissible replacements. The system adopts structured LLM proposals only after deterministic checks. Samples that do not trigger correction retain the original OCR text; rejected or invalid proposals cause a return to that text. The adopted text and decision status are cached for prior encoding.

3.3. Textual, Positional, and Structural Priors

The three prior branches encode character identity, spatial layout, and observed stroke evidence. The textual branch combines adopted character embeddings, one-dimensional order embeddings, confidence embeddings, and a correction-status representation. A lightweight Transformer encoder [18] maps them to 256-dimensional conditions while retaining the sequence axis for visual queries. Validity masks identify padding and empty text; ground-truth transcription length is never used to repair the OCR token mask. The text encoder is fixed across the main experiments and ablations. It processes character conditions, whereas the bottleneck Mamba processes visual features.

The positional branch encodes OCR detection boxes from I_{LR}. Coordinates follow the image's cropping, resizing, and padding transformations before normalization by the current width and height. A rectangular box is represented by its two corners, centre, width, and height; a quadrilateral retains its eight vertex coordinates together with centre and scale information. A multilayer perceptron maps these geometric vectors to 256 dimensions and combines them with relative-order encodings. When only a line box is available, characters share its geometry; interpolated character boxes are not created. Missed detections, overlapping boxes, and unmatched character locations receive explicit validity flags. Text-instance indices preserve correspondence in complete scenes. Coordinate inversion checks compare recovered positions with the original detected boxes and record any crop-induced truncation.

The structural branch computes horizontal and vertical Sobel gradients G_x and G_y from the grayscale or luminance channel of I_{LR}. Gradient magnitude is divided by its spatial maximum plus a numerical stabilizer, and orientation is $\Theta = \mathrm{atan2}(G_y, G_x)$. The image, normalized magnitude, $\sin\Theta$, $\cos\Theta$, and a text-region mask are concatenated along channels. The region mask comes from LR OCR detections; when detection fails, the whole image is valid so that visual structure remains available. A convolutional encoder transforms the LR-derived structural inputs into a 256-channel two-dimensional feature map while preserving their spatial organization. The resulting map is retained for spatial feature extraction and is resized to the required scales when necessary. Before prior fusion, the features associated with valid OCR regions are pooled and projected into the common 256-dimensional token space. The same convolutional structure encoder is used throughout, with no additional visual state-space encoder in this branch. It uses no HR image, transcription, LLM-rendered glyph, or RRDB output. The horizontal and vertical gradients, local orientation cues, and region-aligned structural features provide a spatial representation of the observed stroke geometry. This representation can encode approximately balanced or reflected stroke arrangements when they are present, while also retaining unequal stroke positions, openings, and asymmetric component layouts. The structural encoder does not explicitly identify symmetry axes, pair reflected strokes, impose reflection equivariance, or synthesize missing strokes through geometric reflection. Therefore, it incorporates symmetry-relevant visual evidence without introducing an explicit symmetry constraint.

Each prior branch has an independent projection and validity mask. Before fusion, all available prior representations are aligned to a common token tensor of shape $B \times N \times 256$, where B is the batch size and N denotes the padded number of OCR character or text-instance positions.

Textual and positional tokens follow the same OCR token indices. The structural encoder initially produces a two-dimensional feature map of shape $B \times 256 \times H_s \times W_s$. For each valid OCR region, spatial features are selected according to the corresponding transformed LR bounding-box coordinates and aggregated by region-wise average pooling. The pooled representation is then projected into a 256-dimensional structural token associated with the corresponding OCR index.

When only a text-line bounding box is available, its pooled structural representation is associated with the valid character positions of that line. No artificial character-level bounding boxes or HR-derived structural features are introduced. Validity masks distinguish observed OCR regions from padded or unavailable positions. When OCR detection is unavailable, the structural branch retains its LR-image-derived whole-image fallback independently of the positional branch.

Therefore, the two-dimensional structural feature map and the aligned structural tokens represent successive stages of the same structural prior pathway. The spatial feature map preserves image-derived structural information, whereas the aligned tokens are used for gated prior fusion and subsequent key/value construction in cross-attention. The resulting token-aligned prior is defined as follows:

$$P_{\mathrm{fuse}} = \sum_{m} \alpha_m \, A_m(P_m) \tag{5}$$

The index m identifies the textual, positional, and structural priors. Equation (5) sums their aligned representations at corresponding token positions. The interface retains the independent priors, validity masks, fusion weights, and version metadata. A sample-level reliability vector combines OCR confidence, correction status, the fraction of valid positions, a structural-response summary, and globally pooled visual features. It contains

no ground-truth correctness, HR image, or test-set difficulty label. The gating network maps this vector to branch scores a_m and normalizes them over valid branches:

$$\alpha_m = \frac{v_m \exp(a_m)}{\sum_k v_k \exp(a_k) + \varepsilon_g} \tag{6}$$

Here, v_m indicates branch validity, and ε_g stabilizes the denominator. Invalid branches are excluded before normalization; if every condition is invalid, condition injection is bypassed. Gate training starts from the equal-weight configuration, then learns sample-level weights before final joint fine-tuning. Gate entropy and branch gradients are monitored for persistent collapse, without adding an entropy term to the joint objective. Gating adjusts the relative contribution of available evidence; an unchanged OCR string can still be incorrect.

The normalized gate outputs are relative fusion coefficients for the textual, positional, and structural branches, rather than probabilities that the corresponding priors are correct. A smaller textual coefficient reduces the relative contribution of the textual representation to the fused condition, but does not by itself establish a reduction in reconstruction errors. OCR confidence is one component of the reliability vector, which also includes correction status, the fraction of valid positions, a structural-response summary, and pooled visual features. Therefore, confidence-dependent weight patterns and confidence-dependent reconstruction improvements provide distinct forms of evidence and should be examined separately.

For cross-attention, the same weights act on the aligned branch features before key/value projection. At token position i, the text and position contributions form a partial sum:

$$P_{\mathrm{tp,i}} = \alpha_{\mathrm{txt}} A_{\mathrm{txt}}(P_{\mathrm{txt}})_i + \alpha_{\mathrm{pos}} A_{\mathrm{pos}}(P_{\mathrm{pos}})_i \tag{7}$$

$$P_{\mathrm{cond,i}} = P_{\mathrm{tp,i}} + \alpha_{\mathrm{str}} A_{\mathrm{str}}(P_{\mathrm{str}})_i = P_{\mathrm{fuse,i}} \tag{8}$$

Equation (8) adds the structural contribution at the same token position, so P_{cond} is exactly the fused sequence P_{fuse} from Equation (5). The token axis is preserved through this weighted sum. At injection scale l, the key and value tensors are $K_l = P_{\mathrm{cond}} W_l^K$ and $V_l = P_{\mathrm{cond}} W_l^V$, with independent projections at CA-B and CA-M. Thus, the gate weights enter both attention tensors through the same fused condition sequence. No additional concatenation or token normalization is inserted between this sum and the key/value projections. Invalid token positions are handled by the attention mask.

3.4. Cross-Attention-Guided Residual Diffusion

Residual diffusion refines the RRDB reconstruction and preserves its direct visual recovery path. The forward diffusion process [19] adds Gaussian noise to x_0 at timestep t:

$$x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1 - \bar{\alpha}_t}\varepsilon, \varepsilon \sim \mathrm{N}(0, I) \tag{9}$$

Here, $\bar{\alpha}_t$ is the cumulative product of signal-retention coefficients. The U-Net concatenates x_t and I_{init} along channels and maps them to 64 base channels. Encoder–decoder channel multipliers are 1/2/4/4, and timestep embeddings modulate the residual blocks. The output predicts noise ε_θ during both training and evaluation. At CA-B and CA-M, visual features supply queries Q, while the condition interface supplies keys K and values V. Invalid condition tokens are masked before attention normalization:

$$F' = F + \gamma \mathrm{Proj}\left(\mathrm{softmax}\left(\frac{QK^{\mathrm{T}}}{\sqrt{d}} + M\right)V\right) \tag{10}$$

Equation (10) shows a single attention head. M is the additive validity mask, d is the head dimension, and γ is a learnable residual scale initialized to zero. In the multi-head implementation, Proj concatenates heads, projects the output, and restores spatial dimensions. Both injection sites use eight heads of dimension 32 and receive all three prior types through independent key/value projections. Their source priors, reliability information, and masks are shared. CA-M is located at decoder stage U2 at one-quarter resolution; no cross-attention is applied at the shallow high-resolution U1 stage.

Positional information is incorporated into the conditioning pathway through the OCR bounding-box encoder described in Section 3.3. The detected bounding-box coordinates are transformed consistently with image cropping, resizing, and padding before normalization and feature encoding. The resulting positional tokens are aligned with the corresponding textual and structural tokens and incorporated into the fused conditioning representation.

At both CA-B and CA-M, positional information contributes to the attention key and value tensors through the fused prior sequence. No additional distance-based spatial bias is added directly to the attention logits in the evaluated implementation. Accordingly, Equation (10) contains the query–key similarity term and the additive validity mask, without a separate spatial-bias term.

When OCR detection is missing, the positional branch is marked invalid rather than assigned an artificial full-image bounding box. In contrast, the structural branch retains a whole-image fallback derived from the LR observation. This distinction preserves structural visual evidence without introducing unsupported positional information.

Training uses 1000 diffusion timesteps with a cosine schedule. For inference, denoising diffusion implicit model (DDIM) sampling [20] uses $\eta = 0$ and uniformly spaced selections from the training timesteps: 15 steps for TextZoom and 20 for Real-CE. This setting adds no reverse-process noise, but the reconstruction still depends on the initial noise.

3.5. Mamba-Based Bottleneck Sequence Modeling

The bottleneck module propagates horizontal visual context within each row, without connecting unrelated row boundaries. For F_b of shape $B \times C \times H_b \times W_b$, we arrange each row as a left-to-right sequence and combine the batch and row dimensions to obtain BH_b sequences of length W_b. The processed sequences return to their original spatial positions. The selective state-space update [9] is:

$$h_j = \bar{A}_j h_{j-1} + \bar{B}_j u_j, z_j = C_j h_j + D u_j \quad (11)$$

The index j is the visual sequence position, not the diffusion timestep. $\bar{A}_j$ and $\bar{B}_j$ are discretized state-update parameters, C_j projects the state to the output, and D defines a direct pathway. The state transition uses an input-dependent discretization step; the continuous transition is discretized through its matrix exponential, without requiring an explicit matrix inverse in the implementation. The block applies normalization, input projection, local depthwise convolution, selective scanning, gating, and output projection. It follows bottleneck cross-attention and returns its output through a residual connection whose scale is initialized to zero.

We use one Mamba block with state dimension 16, expansion factor 2, and local convolution kernel size 4. The horizontal sequence lengths for TextZoom and Real-CE training crops are 16 and 32, respectively. The scan is unidirectional and fixed across test samples. It models horizontal context without imposing bidirectional symmetry or reflection equivariance. Its coverage of vertical, curved, and multidirectional text is limited. Figure 3 illustrates the scan and its placement at the bottleneck.

For parameter accounting, the bottleneck Mamba module operates on 256-dimensional visual features with a state dimension of 16, an expansion factor of 2, a local convolution kernel size of 4, and an automatically determined time-step rank of 16. Under the adopted standard Mamba implementation, the core operator contains 437,760 learnable parameters, corresponding to approximately 0.438 M parameters. The surrounding normalization and residual scaling introduce a negligible additional parameter count at the reported precision. Accordingly, introducing the Mamba module increases the total reconstruction-network parameter count from 30.88 M to 31.32 M. These values distinguish the size of the core state-space operator from the parameter count of the complete reconstruction network.

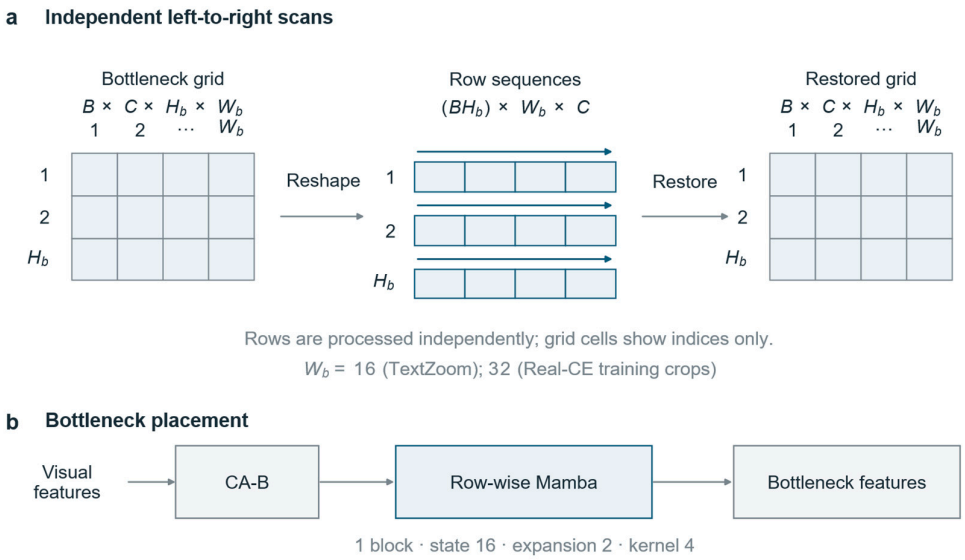


Figure 3. Independent row-wise Mamba processing at the diffusion bottleneck. (**a**) Each row of the bottleneck feature grid forms a left-to-right sequence. Combining batch and row dimensions produces BH_b sequences of length W_b, which are restored to their original grid positions after processing. The small grids illustrate indices only, not measured feature values. (**b**) One row-wise Mamba block follows bottleneck cross-attention (CA-B), with state dimension 16, expansion factor 2, and local convolution kernel size 4. The horizontal sequence lengths for the TextZoom and Real-CE training crops are 16 and 32, respectively. Rows are scanned independently; no bidirectional scan or connection between row ends is used.

3.6. Joint Training Objective

The objective combines diffusion noise prediction with image and text reconstruction supervision:

$$L_{\text{total}} = L_{\text{diff}} + \lambda_{\text{pix}} L_{\text{pix}} + \lambda_{\text{perc}} L_{\text{perc}} + \lambda_{\text{ocr}} L_{\text{ocr}} + \lambda_{\text{edge}} L_{\text{edge}} \quad (12)$$

During training, image-based losses use a single-step clean-residual estimate at the sampled timestep t. Let $\hat{\varepsilon}_t = \varepsilon_\theta(x_t, t, I_{\text{init}}, P_{\text{cond}})$ denote the predicted noise. Inverting Equation (9) gives the residual estimate and the corresponding training image:

$$\hat{x}_0^{(t)} = \frac{x_t - \sqrt{1 - \bar{\alpha}_t}\hat{\varepsilon}_t}{\max(\sqrt{\bar{\alpha}_t}, \delta)} \quad (13)$$

$$I_{\text{SR}}^{(t)} = \text{clip}_{[0,1]}(I_{\text{init}} + \hat{x}_0^{(t)}) \quad (14)$$

The denominator floor stabilizes residual inversion, and clipping constrains the estimated reconstruction to the normalized image range. When the denominator floor is inactive, substituting the residual definition and the forward diffusion equation into the single-step reconstruction expression causes the explicit initialization terms to cancel. The resulting image estimate can therefore be expressed as the HR target plus a timestep-dependent scaled noise-prediction error. Consequently, the pixel and edge reconstruction losses are closely related to the noise-prediction error rather than providing independent direct supervision of the RRDB output.

The corresponding scaling factor increases as the cumulative signal-retention coefficient decreases, potentially amplifying reconstruction errors at high diffusion timesteps. The denominator floor prevents division by arbitrarily small values, while clipping bounds the reconstructed image but does not completely eliminate this amplification effect. These properties are taken into account when interpreting the joint training objective. Multi-step DDIM sampling is used for inference and is not unrolled to compute the training losses.

L_{diff} is the mean squared error between sampled and predicted noise under the residual parameterization. L_{pix} is the masked L_1 distance between I_{SR} and I_{HR}. L_{perc} sums the L_1 differences of frozen VGG19 features at relu1_2, relu2_2, and relu3_4, with each layer normalized by its feature count $C_j H_j W_j$. Both datasets use the same VGG19 layers and preprocessing configuration. LPIPS is used only for evaluation and is not the training perceptual loss.

L_{ocr} is the CTC negative log-likelihood [21] from a frozen recognizer, and its mean uses only samples with valid transcriptions as the denominator; regions without valid transcriptions are masked out rather than entered as empty-string targets. Recognizer parameters stay fixed, while gradients with respect to I_{SR} are retained. L_{edge} sums the masked horizontal and vertical Sobel-gradient L_1 differences between I_{SR} and I_{HR}, normalized by twice the sum of valid-mask values plus a stabilizer. HR-derived gradients serve only as training targets and never enter the structural prior, gate, cross-attention conditions, or inference cache.

The loss weights are set to $\lambda_{\text{pix}} = 1.00$, $\lambda_{\text{perc}} = 0.01$, $\lambda_{\text{ocr}} = 0.10$, and $\lambda_{\text{edge}} = 0.10$ for both datasets. These coefficients balance four complementary objectives: pixel-level fidelity to the target image, perceptual similarity in the frozen VGG19 feature space, recognition-oriented supervision through the fixed OCR recognizer, and preservation of horizontal and vertical image gradients. The pixel loss provides the primary reconstruction constraint, whereas the perceptual, OCR, and edge terms provide complementary feature-level, text-readability, and structural supervision.

The coefficients were selected using internal training-validation splits, considering image quality, recognition performance, and training stability. The official TextZoom test set and final Real-CE evaluation set were excluded from weight selection. The same coefficients were retained across the two datasets to avoid dataset-specific loss tuning and to evaluate a shared training objective under two distinct settings. To examine the sensitivity of this choice, we varied the OCR and edge loss coefficients independently while keeping the pixel and perceptual coefficients fixed. The resulting validation comparisons are reported in Section 4.4. This procedure evaluates the stability of the selected configuration without using the test sets to select loss weights.

HR images and ground-truth transcriptions are used for training supervision and are excluded from inference conditions. RRDB is first pretrained using direct image reconstruction supervision and subsequently frozen during diffusion training. In the final joint fine-tuning stage, the last two RRDBs are unfrozen, while the remaining RRDB parameters and external supervision networks remain fixed.

During joint fine-tuning, the reconstruction losses do not include a separate direct penalty on the initial RRDB output. Nevertheless, the initial reconstruction remains part of the residual construction and the noise predictor's conditioning pathway. Gradients can therefore propagate to the unfrozen RRDBs through these dependencies. This indirect gradient pathway is distinct from direct supervision of the initial reconstruction during RRDB pretraining. Accordingly, the final fine-tuning stage should be interpreted as joint optimization through the residual prediction framework, rather than as an independently supervised refinement of the RRDB initialization.

4. Experiments

We evaluated reconstruction quality, the additional benefit of OCR correction, component contributions, and computational cost. Separate analyses used an independent recognizer, Real-CE language groups, and TextZoom degradation groups.

4.1. Datasets and Evaluation Protocol

We trained separate models on TextZoom [1] and Real-CE [2]. TextZoom combined train1 and train2 into 17,367 training pairs and used 4373 test pairs after the study's data cleaning, with 1619 easy, 1411 medium, and 1343 hard samples. LR and HR images were normalized to 16 × 64 and 32 × 128 pixels for ×2 super-resolution. Real-CE used images acquired at 13 mm as LR inputs and 52 mm as HR targets for ×4 super-resolution. Training used text-aware paired LR 64 × 64 and HR 256 × 256 crops. Final evaluation retained full scenes and used valid_list to select 261 valid scenes: 116 Chinese, 101 English, and 44 mixed Chinese–English scenes.

Image quality was measured by peak signal-to-noise ratio (PSNR), structural similarity index (SSIM) [22], and learned perceptual image patch similarity (LPIPS) [23]. Exact-match accuracy (Acc) required the recognized string to match the reference after the fixed text-normalization procedure. Character error rate (CER) counted Levenshtein insertions, deletions, and substitutions relative to the number of reference characters. Reference transcriptions came from dataset annotations rather than OCR of HR images. After applying the same fixed text-normalization procedure to predictions and references, let p_i and r_i denote the predicted and reference strings for sample i in an evaluation subset g containing M_g samples. Exact-match accuracy (Acc, reported as a percentage) and corpus-level character error rate (CER) are defined as follows:

$$Acc_g = \frac{100}{M_g}\sum_{i=1}^{M_g} 1[p_i = r_i], \quad CER_g = \frac{\sum_{i=1}^{M_g} ED(p_i, r_i)}{\sum_{i=1}^{M_g} |r_i|} \tag{15}$$

Here, $1[\cdot]$ is the indicator function, ED is the Levenshtein edit distance counting character substitutions, deletions, and insertions, and $|r_i|$ is the number of reference characters. CER is the ratio of total character edits to total reference characters within the subset, rather than the mean of per-string normalized edit distances. Acc measures exact-string correctness, whereas CER measures character-level error severity; within a subset, $1 - \mathrm{Acc}/100$ is the string error rate and is not the definition of CER.

For TextZoom, g denotes the easy, medium, or hard subset. Each metric is first computed within each subset, and the overall result is the equal-weight arithmetic mean of the three subset values:

$$Acc = \frac{1}{3}\sum_{g} Acc_g, \quad CER = \frac{1}{3}\sum_{g} CER_g \tag{16}$$

The same equal-weight macro-averaging procedure is applied to PSNR, SSIM, LPIPS, exact-match accuracy (Acc), and character error rate (CER). Specifically, the TextZoom

test set consists of three difficulty subsets: Easy (1619 samples), Medium (1411 samples), and Hard (1343 samples). Each evaluation metric is first calculated independently within each subset, and the overall benchmark result is obtained as the unweighted arithmetic mean of the three subset-level values, as defined in Equation (16).

It is important to distinguish this macro-averaging procedure from pooled evaluation over all 4373 test samples. Since the three difficulty subsets contain different numbers of samples, the unweighted macro-average and the corresponding pooled result are not necessarily identical. In particular, corpus-level CER is calculated as the total number of character-level edit operations divided by the total number of reference characters within each subset. For pooled evaluation, CER is computed directly from the total character edits and reference characters over the complete test set, rather than by weighting subgroup CER values according to their sample counts.

To improve the transparency of the evaluation protocol, the individual Easy, Medium, and Hard results are additionally reported in Section 4.2. The benchmark macro-averaged results and the pooled statistics used in the degradation analysis are explicitly distinguished throughout the experiments.

For Real-CE, image metrics were aggregated over all valid aligned regions, and recognition metrics over text instances with valid reference transcriptions. Empty annotations and regions without valid transcriptions were excluded from recognition denominators. The same valid region set was used for both recognizers. Overall results were calculated from all valid evaluation units, rather than by averaging the three language groups or weighting them by scene counts. PaddleOCR was the primary recognizer; its weights, character normalization, preprocessing, and decoding rules were fixed across methods. Frozen EasyOCR was used only after reconstruction models had been trained and did not participate in prior generation, OCR-loss supervision, checkpoint selection, or tuning.

Because PaddleOCR is used both to construct OCR-derived conditions and to provide recognition-oriented training supervision, evaluation with the same frozen recognizer may exhibit recognizer-specific advantages. We therefore distinguish the primary PaddleOCR evaluation from independent recognition evaluation with EasyOCR. EasyOCR is applied only to completed reconstruction outputs, without access to prior generation, training gradients, checkpoint selection, or hyperparameter tuning. The same reference transcriptions, text-normalization rules, and valid evaluation units are used when comparing the two recognizers. The independent evaluation is intended to test whether the observed reconstruction gains persist under a different recognition model, rather than to establish complete independence of the training and evaluation framework.

All training and testing used a single NVIDIA RTX 4090 (24 GB), with Ubuntu 22.04, Python 3.10, PyTorch 2.1.0, and CUDA 12.1. AdamW used an initial learning rate of 1×10^{-4} with cosine decay to 1×10^{-6} (Table 1). RRDB was pretrained for 100,000 steps, and the checkpoint with the highest internal-validation PSNR was retained as B0. RRDB was then frozen while the diffusion network, prior encoders, cross-attention, Mamba, and gating modules were trained. Finally, the last two RRDBs were unfrozen for 10,000 joint fine-tuning steps at 1×10^{-5}. Batch size/gradient accumulation were 8/2 for TextZoom and 2/4 for Real-CE. Loss weights were selected only on an internal split of the training data; the TextZoom test set and final Real-CE evaluation set were excluded from tuning.

Table 1. Principal implementation settings.

Item	Setting
OCR and LLM	PaddleOCR; Top-K = 5; calibrated OCR temperature T = 1.18; sample-confidence threshold = 0.65; candidate-gap threshold = 0.20; maximum edit budget D_max = 2; correction-stage scoring coefficients λ_lm = 1.0, λ_ocr = 1.0, and λ_ed = 0.5; Qwen2.5-7B-Instruct; BF16; greedy decoding; do_sample = False; num_beams = 1; max_new_tokens = 128; repetition_penalty = 1.0; structured JSON output; teacher-forced candidate scoring; offline caching.
Priors and cross-attention	Condition dimension: 256; injection at CA-B and CA-M; 8 heads with 32 dimensions per head
Diffusion backbone	U-Net base channels: 64; channel multipliers: 1/2/4/4; 1000-step cosine schedule
Inference sampling	DDIM, η = 0; 15 steps for TextZoom and 20 for Real-CE
Mamba	1 bottleneck block; state dimension: 16; expansion factor: 2; local convolution kernel: 4
Optimizer and learning rate	AdamW; initial rate: 1×10^{-4}; cosine decay to 1×10^{-6}
Hardware and software	RTX 4090 24 GB; Ubuntu 22.04; PyTorch 2.1.0; CUDA 12.1
Training stages	RRDB: 100,000 steps; frozen RRDB during diffusion training; last 2 RRDBs fine-tuned for 10,000 steps at 1×10^{-5}
Batch size/accumulation	TextZoom: 8/2; Real-CE: 2/4
Repeated ablations	5 seeds: 2022, 2023, 2024, 2025, 2026; paired within each independently trained ablation group

4.2. Quantitative Comparison with Baseline Methods

Table 2 compares conventional image super-resolution networks and text-oriented reconstruction methods on TextZoom. Table 3 compares RRDB and the proposed full model across the three TextZoom difficulty subsets, and Table 4 reports the comparison of methods on Real-CE. Since the compared methods were originally developed under different training and evaluation settings, their numerical results must be interpreted with respect to the corresponding experimental protocols.

The baseline comparison distinguishes results obtained through direct computation, previously reported evaluations, and models trained or evaluated in the present study. Particular attention is given to the implementation sources, model checkpoints, super-resolution scale factors, input preprocessing, and evaluation procedures. The previously reported results are considered reference measurements unless equivalent evaluation conditions can be established. Consequently, differences from the original publications are examined explicitly rather than attributed solely to the choice of evaluation protocol.

For Real-CE, additional consideration is given to the distinction between whole-image reconstruction methods and text-line-based networks. The latter may require text-region extraction, resizing, padding, or scale-factor adaptation to process scene images. The actual preprocessing and evaluation procedures adopted for these methods are described separately to clarify the scope and comparability of the reported results.

Table 2. Quantitative comparison on TextZoom under the evaluation protocol used in this study.

Method	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓	Acc ↑ (%)	CER ↓
Bicubic	20.52	0.666	0.345	58.3	0.335
SRCNN [24]	21.73	0.744	0.310	64.9	0.298
ESRGAN [16]	22.66	0.781	0.259	69.2	0.267
SwinIR [25]	23.76	0.823	0.211	76.4	0.225
TATT [3]	24.24	0.842	0.196	79.8	0.198
TPGSR [4]	24.18	0.846	0.190	80.5	0.191

DPMN [5]	24.40	0.863	0.184	81.9	0.180
TextSR [12]	24.71	0.871	0.179	82.5	0.174
PRISM [13]	24.88	0.878	0.176	82.9	0.170
B0: RRDB [16]	23.62	0.817	0.224	72.4	0.238
Full model	25.06	0.883	0.173	83.4	0.164

Table 2 summarizes the quantitative results associated with the TextZoom evaluation. Each overall metric is calculated using equal-weight macro-averaging across the Easy, Medium, and Hard subsets. Recognition evaluation follows the fixed PaddleOCR configuration and text-normalization protocol described in Section 4.1. The TATT and DPMN values coincide with the corresponding results reported in [11]; their relationship to the earlier evaluation is discussed in Section 5.4. Cross-study numerical comparisons are interpreted with consideration of the corresponding evaluation settings. ↑ indicates higher is better; ↓ indicates lower is better.

Table 3 presents the reconstruction and recognition performance of RRDB and the proposed full model across the three TextZoom difficulty subsets. Both methods exhibit decreasing performance as the difficulty increases, reflecting the greater challenges associated with more severe image degradation. The proposed method achieves PSNR values of 27.12, 25.24, and 22.82 dB on the Easy, Medium, and Hard subsets, respectively, with corresponding exact-match accuracies of 91.9%, 83.8%, and 74.5%.

Table 3. Difficulty-stratified quantitative evaluation of RRDB and the proposed full model on TextZoom.

Model	Difficulty	PSNR ↑	SSIM ↑	LPIPS ↓	Acc (%) ↑	CER ↓
RRDB	Easy	25.85	0.865	0.164	82.1	0.142
RRDB	Medium	23.78	0.815	0.222	72.1	0.226
RRDB	Hard	21.23	0.771	0.286	63.1	0.346
RRDB	Overall	23.62	0.817	0.224	72.4	0.238
Full model	Easy	27.12	0.921	0.115	91.9	0.072
Full model	Medium	25.24	0.882	0.168	83.8	0.157
Full model	Hard	22.82	0.846	0.236	74.5	0.263
Full model	Overall	25.06	0.883	0.173	83.4	0.164

The Easy, Medium, and Hard subsets contain 1619, 1411, and 1343 samples, respectively. Overall results are calculated as the unweighted arithmetic mean of the three subset-level metrics according to Equation (16). The same evaluation protocol and recognition configuration are applied to both models. ↑ indicates higher is better; ↓ indicates lower is better.

Compared with RRDB, the proposed full model improves PSNR by 1.27, 1.46, and 1.59 dB on the three subsets, respectively. The corresponding improvements in exact-match accuracy are 9.8, 11.7, and 11.4 percentage points. Furthermore, CER decreases from 0.142 to 0.072 on Easy, from 0.226 to 0.157 on Medium, and from 0.346 to 0.263 on Hard. These results indicate consistent improvements in image reconstruction quality and character-level recognition across the evaluated difficulty levels.

Following the equal-weight macro-averaging procedure defined in Equation (16), the overall PSNR, SSIM, LPIPS, exact-match accuracy, and CER of the proposed model are 25.06 dB, 0.883, 0.173, 83.4%, and 0.164, respectively. These values agree with the corresponding full-model results in Table 2. The explicit difficulty-stratified evaluation therefore makes the calculation of the reported overall benchmark metrics transparent and directly verifiable.

On TextZoom, the full model achieved 25.06 dB PSNR, 0.883 SSIM, 0.173 LPIPS, 83.4% Acc, and a CER of 0.164. Compared with RRDB, the full model improved PSNR by 1.44 dB and Acc by 11.0 percentage points, while reducing CER by 0.074. Relative to DPMN,

the corresponding gains were 0.66 dB in PSNR and 1.5 percentage points in Acc, with a CER reduction of 0.016.

We further compared the proposed method with TextSR and PRISM. TextSR achieved 24.71 dB PSNR, 0.871 SSIM, 0.179 LPIPS, 82.5% Acc, and a CER of 0.174, whereas PRISM achieved 24.88 dB PSNR, 0.878 SSIM, 0.176 LPIPS, 82.9% Acc, and a CER of 0.170. Relative to TextSR, the full model improved PSNR by 0.35 dB, SSIM by 0.012, and Acc by 0.9 percentage points, while reducing LPIPS by 0.006 and CER by 0.010. Relative to PRISM, the corresponding differences were +0.18 dB in PSNR, +0.005 in SSIM, and +0.5 percentage points in Acc, with reductions of 0.003 in LPIPS and 0.006 in CER.

These comparisons indicate that the full model achieved the best reported values among the listed methods for the five evaluation metrics under the stated protocol.

Table 4. Quantitative comparison on Real-CE under the evaluation protocol used in this study.

Method	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓	Acc ↑ (%)	CER ↓
Bicubic	20.01	0.689	0.368	28.1	0.458
TSRN [1]	22.67	0.765	0.286	43.8	0.329
TATT [3]	23.12	0.779	0.268	45.3	0.319
TPGSR [4]	23.62	0.791	0.252	47.2	0.307
PEAN [7]	24.05	0.804	0.235	49.1	0.294
B0: RRDB [16]	23.91	0.797	0.245	47.8	0.305
Full model	24.38	0.815	0.223	50.2	0.283

All baseline results in this table were obtained by the authors through direct computation or method reproduction and evaluation. The evaluation set contains 261 valid scenes. Image-quality metrics are aggregated over valid aligned image regions, and recognition metrics over text instances with valid reference transcriptions. Overall results are calculated from the valid evaluation units rather than by averaging the Chinese, English, and mixed Chinese–English group scores. ↑ indicates higher is better; ↓ indicates lower is better. For the Bicubic baseline, PaddleOCR is evaluated on the bicubically upsampled images at the target reconstruction resolution. Its recognition accuracy is not computed from the cached OCR transcription of the original LR image used for prior construction.

On Real-CE, the full model increased PSNR by 0.47 dB and Acc by 2.4 percentage points over RRDB, while reducing CER by 0.022. Compared with PEAN, PSNR and Acc increased by 0.33 dB and 1.1 percentage points, respectively. The gains over RRDB were smaller than those on TextZoom. Because the datasets differ in scale factor, image composition, and evaluation units, their absolute scores should not be interpreted as measurements under equivalent task conditions. Figure 4 summarizes the PSNR and Acc comparisons.

4.3. OCR Correction and Its Effect on Reconstruction

Constrained correction improved the input text predictions before image reconstruction (Table 5). On TextZoom, Acc increased from 58.3% to 60.1%, and CER decreased from 0.326 to 0.301. On Real-CE, Acc increased from 44.7% to 46.3%, and CER decreased from 0.332 to 0.316. The miscorrection rates among initially correct OCR predictions were 0.7% and 0.9%, respectively. These measurements describe the OCR correction stage separately from recognition of super-resolved images.

The distinction between OCR-stage accuracy and reconstruction-stage recognition accuracy is particularly important for Real-CE. In Table 5, the raw OCR accuracy of 44.7% (CER = 0.332) refers to the OCR transcriptions extracted directly from the original LR observations and stored for subsequent prior construction. In contrast, the bicubic accuracy

of 28.1% (CER = 0.458) in Table 4 is obtained by applying the frozen evaluation recognizer to images produced by bicubic upsampling to the target resolution.

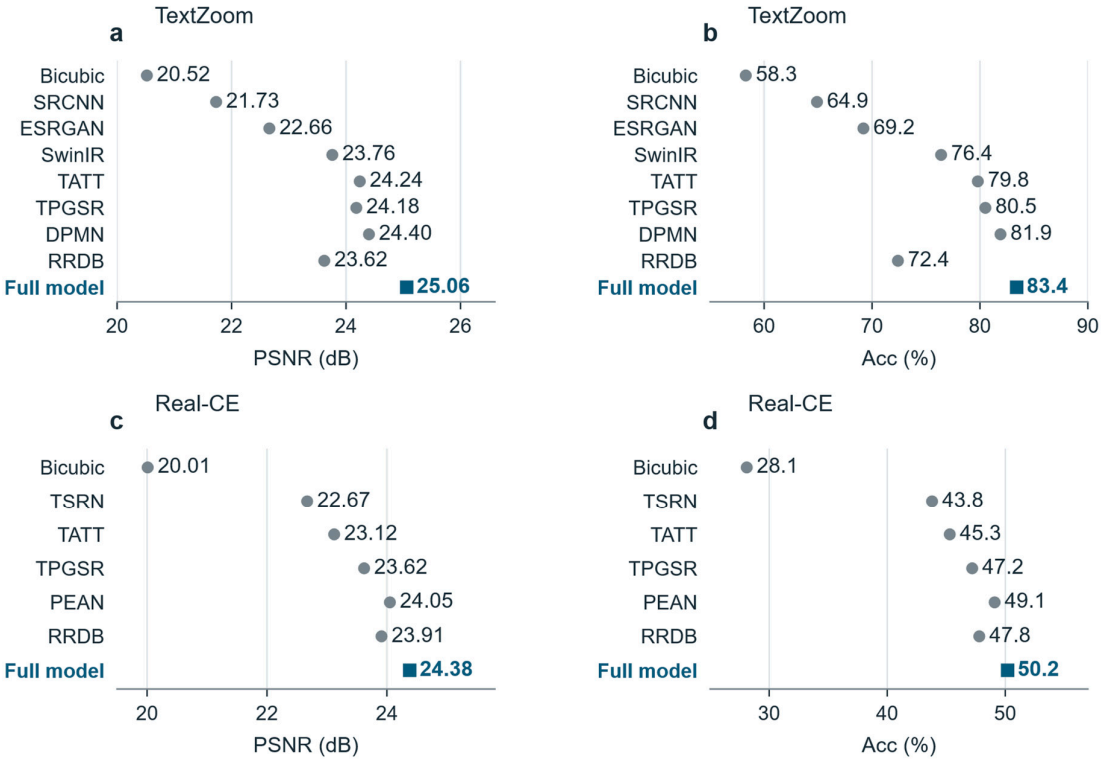


Figure 4. Reconstruction quality and text recognition under the study protocols. (**a**,**b**) TextZoom peak signal-to-noise ratio (PSNR) and exact-match accuracy (Acc); (**c**,**d**) the same metrics on Real-CE. Separate models are trained for TextZoom at ×2 and Real-CE at ×4. Points show selected methods from Tables 2 and 4; blue squares identify the full model. TextSR and PRISM are included in the complete TextZoom comparison in Table 2. TextZoom scores are equal-weight averages over the three difficulty subsets. Real-CE metrics are aggregated over valid aligned regions and transcribed text instances within 261 scenes. Recognition uses fixed PaddleOCR. No uncertainty intervals are shown.

These values therefore correspond to different image inputs and separate recognition passes, rather than two evaluations of an identical OCR output. Image rescaling and interpolation can affect the text detection and recognition pipeline, even though bicubic interpolation does not recover new character information. Nevertheless, the present results do not isolate the respective effects of image rescaling, text detection, and character recognition on this difference. Accordingly, the 16.6-percentage-point discrepancy is reported as an observed difference between the two processing pathways and should not be interpreted as evidence that lower-resolution input is intrinsically more suitable for OCR. The reconstruction comparisons in Table 4 remain based on the designated evaluation procedure applied to the reconstructed outputs.

Table 5. Separate evaluation of OCR predictions and constrained LLM correction.

Dataset	Stage	Acc ↑ (%)	CER ↓	Miscorrection (%)	Rejection (%)	Fallback (%)
TextZoom	Raw OCR	58.3	0.326	—	—	—
TextZoom	Constrained correction	60.1	0.301	0.7	12.1	3.6

Real-CE	Raw OCR	44.7	0.332	—	—	—
Real-CE	Constrained correction	46.3	0.316	0.9	13.9	4.2

Miscorrection uses initially correct OCR strings as the denominator. Rejection is failure of deterministic proposal checks; fallback is a parsing or execution exception. Both use all evaluated samples as denominators. A dash denotes a stage without correction. Raw OCR refers to the frozen recognizer's predictions on the original LR inputs before image reconstruction. The corrected-text results are evaluated at this same prior-construction stage. These OCR-stage values are distinct from the recognition measurements obtained after applying super-resolution or bicubic upsampling.

Rejection denoted a proposal rejected by the deterministic constraints; fallback denoted a parsing or execution exception, such as an unparseable response, timeout, unavailable model, or empty output. Both rates used all evaluated samples as their denominator. Miscorrection counted initially correct OCR strings that became incorrect after an accepted edit, with initially correct strings as the denominator. These definitions keep harmful edits separate from rejected proposals and execution failures. The experiment did not isolate the effect of each constraint, and text-level miscorrection did not measure incorrect characters generated in SR images.

For the P-series ablations, the base conditioning interface denotes a common prior-encoding and feature-injection pathway used to evaluate different combinations of textual, positional, and structural information. Each available prior is projected into a 256-dimensional representation, and valid prior branches are combined using equal-weight averaging. If only one prior is available, its fusion weight is one; if all prior branches are invalid, condition injection is bypassed. The fused condition is aligned with the bottleneck visual features, concatenated along the channel dimension, and passed through a learned 1×1 projection. This base conditioning interface does not use attention-based key/value retrieval, bottleneck Mamba modeling, or the final sample-level dynamic reliability gate.

For the P1-O/P1-L comparison, the correction-status embedding is fixed to the same neutral state in both configurations so that the adopted character sequence is the only changing conditioning input. This differs from the complete-model correction ablation, in which enabling the correction procedure also updates the corresponding correction-status information. In contrast to the base interface, the enhanced cross-attention configuration uses CA-B at the U-Net bottleneck and CA-M at the one-quarter-resolution decoder stage, as defined in Section 3.4. Both modules employ eight attention heads with a head dimension of 32 and receive validity-masked, fused prior tokens through their respective key/value projections. The complete reconstruction framework additionally incorporates sample-level dynamic reliability gating and row-wise Mamba sequence modeling.

The contribution of constrained OCR correction to image reconstruction was first evaluated using the P1-O and P1-L configurations in Table 6. Both configurations employed the same base reconstruction network and text-conditioning interface, with the source of the textual prior being the only experimental difference. P1-O used the original OCR transcription, whereas P1-L adopted the transcription obtained after constrained LLM correction.

To assess the stability of the observed improvement, this comparison was repeated using five paired random seeds (2022–2026). For each seed, the two configurations shared the same initialization strategy, training data, optimization settings, training duration, and checkpoint-selection procedure. Replacing raw OCR text with corrected text increased the mean PSNR from 24.44 to 24.52 dB and exact-match accuracy from 79.7% to 80.4%. The corresponding mean paired improvements were 0.08 dB (95% CI: [0.060, 0.100]) and 0.7 percentage points (95% CI: [0.504, 0.896]), respectively.

These results demonstrate a positive and statistically supported effect of constrained OCR correction within the evaluated base text-conditioned configuration. Since the P1

configurations exclude some components of the complete reconstruction framework, additional full-model comparisons on TextZoom and Real-CE were conducted to assess whether the effect persists in the complete architecture.

Table 6. Ablation of text sources and prior combinations on TextZoom.

Config	Text Prior	Position	Structure	PSNR ↑ (dB)	Acc ↑ (%)
P0	×	×	×	24.18	76.9
P1-O	Raw OCR	×	×	24.44	79.7
P1-L	Corrected text	×	×	24.52	80.4
P2	Corrected text	✓	×	24.64	81.3
P4	×	✓	✓	24.58	80.6
P5	Corrected text	×	✓	24.72	81.9
P3	Corrected text	✓	✓	24.78	82.2

All P-series configurations use the base conditioning interface defined in Section 4.3, including equal-weight fusion of valid prior branches and bottleneck concatenation followed by a learned projection. The configurations differ in the prior types or textual sources indicated in the table. For the P1-O/P1-L comparison, correction-status conditioning is fixed to the same neutral value. Their results are reported as means over five paired random seeds (2022–2026), while the remaining P-series configurations retain their original evaluation settings. The P-series configurations exclude enhanced cross-attention, bottleneck Mamba sequence modeling, and final dynamic reliability gating. ✓ and × indicate presence and absence, respectively.

The ablation results in Table 6 help distinguish the contributions of the textual, positional, and structural priors within the base conditioning framework. Adding corrected text to the configuration without priors (P0 to P1-L) increased PSNR from 24.18 to 24.52 dB and Acc from 76.9% to 80.4%. Replacing raw OCR text with corrected text (P1-O to P1-L), while keeping the base network and conditioning interface unchanged, yielded an additional gain of 0.08 dB in PSNR and 0.7 percentage points in Acc.

The positional prior provided an additional benefit when combined with corrected text: P2 improved PSNR from 24.52 to 24.64 dB and Acc from 80.4% to 81.3% relative to P1-L. Similarly, adding the structural prior to the corrected text (P1-L to P5) increased PSNR by 0.20 dB and Acc by 1.5 percentage points. When the position prior was added to the corrected-text-plus-structure configuration (P5 to P3), PSNR increased from 24.72 to 24.78 dB and Acc from 81.9% to 82.2%.

Taken together, these comparisons indicate that textual, positional, and structural information provide complementary benefits within the tested base conditioning framework. The magnitude of each observed gain depends on the combination in which the prior is introduced; these differences should not be interpreted as independent additive effects. P3 uses all three prior types but does not include the enhanced cross-attention, bottleneck Mamba, or final dynamic gating used in the full model. Its results therefore characterize prior combinations rather than the performance of the complete architecture.

To further investigate the contribution of constrained OCR correction within the complete reconstruction framework, we conducted additional controlled ablation experiments on TextZoom and Real-CE. For each dataset, two full-model configurations were compared: one conditioned on raw OCR transcriptions and the other conditioned on the adopted transcriptions after constrained LLM correction. All remaining components, including the positional and structural priors, cross-attention modules, Mamba-based bottleneck sequence modeling, and dynamic reliability gating, were retained. The same dataset partitions, initialization strategy, optimization settings, diffusion sampling

configuration, and evaluation procedures were used. Each comparison was repeated with five paired random seeds (2022–2026).

On TextZoom, the full model using raw OCR text achieved a mean PSNR of 24.99 dB and an exact-match accuracy of 82.8%. When the raw transcription was replaced with constrained-correction text, these values increased to 25.06 dB and 83.4%, respectively. The mean paired improvements were 0.07 dB in PSNR (95% CI: [0.050, 0.090]) and 0.6 percentage points in recognition accuracy (95% CI: [0.404, 0.796]).

On Real-CE, the corresponding full-model comparison showed that using corrected OCR text increased the mean PSNR from 24.31 to 24.38 dB and exact-match accuracy from 49.6% to 50.2%. The mean paired improvements were 0.07 dB in PSNR (95% CI: [0.034, 0.106]) and 0.6 percentage points in recognition accuracy (95% CI: [0.283, 0.917]).

The paired confidence intervals exclude zero for both metrics on both datasets. These results provide evidence that constrained OCR correction improves downstream reconstruction not only in the reduced text-conditioned configuration but also within the complete framework. Although the absolute improvements are modest, the effect is consistently positive under the evaluated paired-seed settings. Importantly, the P1-O/P1-L comparison isolates the adopted textual source under a fixed correction-status condition, whereas the paired full-model comparisons evaluate the combined effect of corrected text and its associated correction-status information while retaining the same reconstruction architecture and training settings. These results distinguish the downstream contribution of the correction interface from the text-level improvements reported in Table 5, but do not isolate the contribution of each correction-related signal individually.

4.4. Conditioning and Bottleneck Sequence Modeling

With the prior content and all other settings unchanged, cross-attention increased PSNR by 0.08 dB and Acc by 0.4 percentage points over concatenation followed by projection. Table 7 compares three alternatives at the same bottleneck location. Adding the Mamba bottleneck module to the configuration without sequence modeling increased PSNR by 0.12 dB and exact-match accuracy by 0.6 percentage points. The additional parameter count was approximately 0.44 M, while online reconstruction latency increased by 5.40 ms per image.

Table 7. Performance and resource cost of bottleneck sequence modules.

Sequence Module	PSNR ↑ (dB)	Acc ↑ (%)	Parameters (M)	Peak Training Reserved Memory (GB)	Latency (ms)
No sequence block	24.94 ± 0.03	82.8 ± 0.20	30.88	12.4	147.68
Lightweight Transformer	25.02 ± 0.02	83.2 ± 0.16	32.44	13.8	160.90
Lightweight Mamba	25.06 ± 0.04	83.4 ± 0.27	31.32	13.0	153.08

PSNR and Acc are reported as means ± standard deviations over five runs. Peak training reserved memory was measured using the PyTorch CUDA caching allocator under the TextZoom training configuration, with a micro-batch size of 8 and gradient accumulation over 2 steps. These memory values represent the complete training configurations rather than single-image inference memory. Reconstruction latency was measured separately on an NVIDIA RTX 4090 at batch size 1 and excludes offline OCR–LLM preprocessing. Parameter counts include the instantiated reconstruction networks but exclude the external OCR models and LLMs.

Compared with the lightweight Transformer, the Mamba configuration achieved numerically higher mean PSNR and exact-match accuracy by 0.04 dB and 0.2 percentage points, respectively. The complete Mamba-based reconstruction network contained 31.32 M parameters, compared with 32.44 M for the Transformer-based configuration,

corresponding to a reduction of 1.12 M parameters. Reconstruction latency also decreased from 160.90 to 153.08 ms per image, representing a reduction of 7.82 ms.

The recorded peak training reserved memory was 13.0 GB for the Mamba configuration and 13.8 GB for the Transformer configuration. These measurements reflect the memory behavior of the complete training configurations, including intermediate activations, gradients, optimizer states, and temporary computational workspaces. Therefore, the observed 0.8 GB difference should not be attributed solely to the bottleneck feature tensors or interpreted as an intrinsic memory saving of the Mamba operator.

Furthermore, the paired 95% confidence intervals for the PSNR and recognition-accuracy differences included zero. Thus, the observed numerical differences do not establish a statistically significant reconstruction-quality advantage of Mamba over the lightweight Transformer at the two-sided 5% level. Under the evaluated implementation, Mamba achieved comparable reconstruction quality with fewer network parameters and shorter measured reconstruction latency. These findings are specific to the tested configuration and do not establish general advantages across arbitrary sequence lengths, scanning strategies, or hardware platforms.

Under the TextZoom difficulty-subset macro-averaging protocol, dynamic reliability gating increased PSNR from 25.03 to 25.06 dB and exact-match accuracy from 83.1% to 83.4%, corresponding to gains of 0.03 dB and 0.3 percentage points, respectively. Online latency increased from 152.99 to 153.08 ms. Although the aggregate improvements were modest, the results motivated a more detailed examination of the gate under different OCR-confidence conditions.

To examine confidence-dependent behavior, we analyzed the fixed-weight and dynamic-gating configurations using the same TextZoom test samples and evaluation protocol. Based on pre-correction OCR confidence, the 4373 samples were divided into a low-confidence group containing 1530 samples and a higher-confidence group containing 2843 samples.

In the low-confidence group, dynamic gating increased PSNR from 22.800 to 22.860 dB and exact-match accuracy from 72.4% to 73.2%, corresponding to improvements of 0.060 dB and 0.8 percentage points, respectively. In the higher-confidence group, PSNR increased from 26.433 to 26.448 dB and exact-match accuracy from 89.6% to 89.7%, corresponding to improvements of 0.015 dB and 0.1 percentage points.

When the two confidence groups were pooled according to their sample counts, the fixed-weight configuration achieved an overall PSNR of 25.162 dB and an exact-match accuracy of 83.6%, whereas the dynamic-gating configuration achieved 25.193 dB and 83.9%, respectively. These pooled statistics are consistent with the complete-test-set results derived from the difficulty-stratified evaluation in Table 3. They differ from the difficulty-subset macro-averaged results because the two aggregation procedures use different weighting rules.

The larger improvements in the low-confidence group are consistent with the intended behavior of assigning relatively less weight to textual evidence and more weight to structural evidence when OCR confidence is low. The confidence-dependent weight patterns were also consistent with this behavior: the mean textual and structural weights were 0.24 and 0.47 in the low-confidence group, compared with 0.45 and 0.24 in the higher-confidence group. The mean positional weights were 0.29 and 0.31, respectively. The three normalized branch weights summed to 1.00 in each group.

We further conducted a diagnostic analysis on samples with incorrect adopted textual priors and valid multimodal prior alignment. Specifically, a sample was included when its adopted OCR–LLM transcription did not match the normalized reference transcription, the textual conditioning tokens were valid, the corresponding OCR detection region could be aligned with the positional prior, and region-associated structural

features were available without relying solely on the whole-image fallback. Applying these eligibility criteria yielded a subset of 612 TextZoom samples. This subset represents incorrectly conditioned samples with valid textual, positional, and structural prior inputs, rather than all samples with incorrect OCR–LLM transcriptions in the complete test set.

Within this subset, dynamic gating increased PSNR from 22.31 to 22.39 dB and exact-match accuracy from 54.6% to 55.4%, corresponding to improvements of 0.08 dB and 0.8 percentage points relative to fixed global weights. These results suggest that reliability-aware fusion can provide a modest benefit when the adopted textual prior is incorrect but complementary positional and structural evidence remains available. Ground-truth transcriptions were used only for post-inference subset identification and evaluation; they were not provided to the correction or reconstruction models. The subset was defined by transcription correctness and prior validity, not by reconstruction performance or the magnitude of the observed gating improvement.

Taken together, these analyses show that the dynamic gate changed the relative contribution of the prior branches and that the dynamic-gating configuration achieved higher average PSNR and Acc in both confidence groups and in the subset with incorrect adopted text. These results support a modest benefit under the tested conditions, but they do not establish that every unreliable prior is suppressed or that every sample benefits from gating. The observed differences should be interpreted within the evaluated model configuration and protocol.

Cross-attention, sequence-block, and fusion comparisons were repeated as three independently trained ablation groups. Each group used seeds 2022, 2023, 2024, 2025, and 2026. Within a group, variants were paired by seed and shared the data split, optimizer, learning-rate schedule, training length, and checkpoint-selection rule. Because the groups were trained separately, structurally identical full-model configurations could have different standard deviations across groups. Table 8 reports the mean paired differences and their two-sided 95% confidence intervals for the three module comparisons. Each comparison was repeated using five paired random seeds (2022–2026), with identical training and evaluation settings within each seed pair. For a given metric, the paired difference was calculated separately for each seed, and the overall effect was summarized by the mean of these five differences.

Because only five paired runs were available, confidence intervals were constructed using Student's t-distribution rather than the standard normal approximation. Specifically, the two-sided 95% critical value was 2.776 for four degrees of freedom. The confidence interval was calculated as the mean paired difference plus or minus the critical value multiplied by the standard error of the paired differences. This approach accounts for seed-wise pairing and uses the variability of the paired differences rather than treating the two model configurations as independent samples.

After applying the paired t-distribution procedure, the confidence intervals for cross-attention and dynamic reliability gating remained above zero for both PSNR and recognition accuracy. In contrast, the intervals for Mamba relative to the lightweight Transformer included zero for both metrics. Therefore, the numerically higher mean reconstruction quality of Mamba should not be interpreted as a statistically significant advantage at the two-sided 5% level. Figure 5 summarizes the TextZoom results for raw versus corrected OCR text (panels a and b) and compares the bottleneck configurations in terms of PSNR and reconstruction latency (panels c and d).

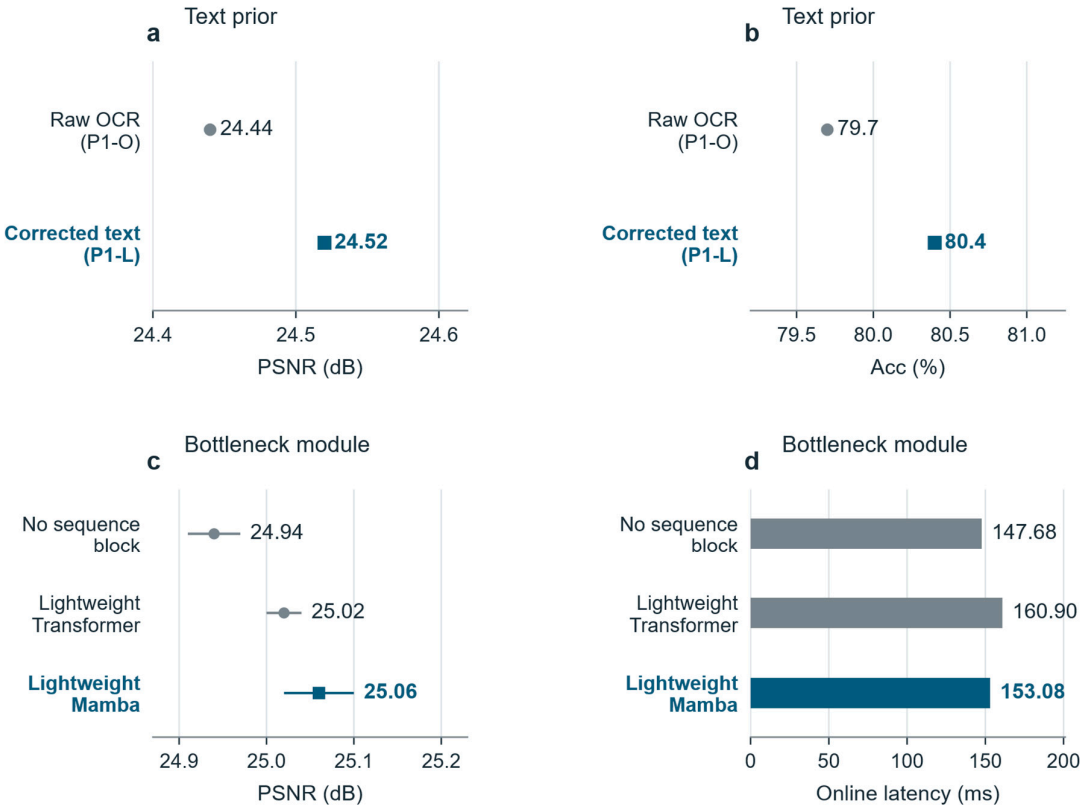


Figure 5. Text-prior correction and bottleneck modeling on TextZoom. (**a**,**b**) Mean PSNR and exact-match accuracy over five paired runs for raw OCR text (P1-O) and constrained-correction text (P1-L) under the same base conditioning interface (Table 6). (**c**,**d**) Mean PSNR and reconstruction latency in the separate bottleneck-module comparison (Table 7). PSNR results for the bottleneck-module comparison are reported as mean ± standard deviation over five runs in Table 7. Timing was measured on an RTX 4090 at batch size 1 and excludes offline OCR–LLM processing. The P1 configurations exclude enhanced cross-attention, bottleneck Mamba, and the final dynamic gate; their results are analyzed separately from the complete-model correction experiments.

Table 8. Paired differences and 95% confidence intervals for key module comparisons.

Comparison	ΔPSNR (dB)	95% CI	ΔAcc (pp)	95% CI
Cross-attention − concatenation/projection	+0.08	[0.025, 0.135]	+0.4	[0.117, 0.683]
Mamba − lightweight Transformer	+0.04	[−0.005, 0.085]	+0.2	[−0.079, 0.479]
Dynamic gating − fixed weights	+0.03	[0.003, 0.057]	+0.3	[0.021, 0.579]

pp denotes percentage points. Each comparison used five paired random seeds (2022–2026). The two-sided 95% confidence intervals were calculated from the seed-wise paired differences using Student's t-distribution with four degrees of freedom (t = 2.776). Specifically, each interval was computed as the mean paired difference ± 2.776 times its standard error. The three ablation groups were trained independently, with variants paired by seed within each group. Confidence intervals containing zero were not interpreted as evidence of statistically significant differences at the two-sided 5% level.

To examine the sensitivity of the joint training objective to the loss weights, we varied the OCR and edge loss coefficients independently around the default configuration while keeping the pixel and perceptual coefficients fixed. The OCR coefficient was evaluated at

0.05, 0.10, and 0.20 with the edge coefficient fixed at 0.10; the edge coefficient was evaluated at 0.05, 0.10, and 0.20 with the OCR coefficient fixed at 0.10. The five configurations were evaluated on the same internal validation split using the same image-quality and recognition metrics. The results are reported as mean ± standard deviation over five runs.

As shown in Table 9, the default configuration achieved 25.11 ± 0.04 dB PSNR, 0.885 ± 0.001 SSIM, 0.170 ± 0.002 LPIPS, 83.7 ± 0.3% Acc, and 0.166 ± 0.003 CER. Reducing the OCR loss weight to 0.05 lowered Acc to 82.9 ± 0.4% and increased CER to 0.174 ± 0.004. Increasing the OCR loss weight to 0.20 improved Acc to 83.9 ± 0.4% and reduced CER to 0.164 ± 0.003, although PSNR remained below the default result at 25.07 ± 0.05 dB.

The edge-weight comparison showed a complementary trade-off. Reducing the edge loss weight to 0.05 produced lower PSNR and SSIM and higher LPIPS than the default configuration. Increasing it to 0.20 improved PSNR to 25.08 ± 0.04 dB, SSIM to 0.884 ± 0.001, and LPIPS to 0.172 ± 0.002, while Acc was slightly lower than the default value at 83.6 ± 0.3%. These results indicate that the loss weights affect image fidelity and text recognition differently. The default setting provides a balanced configuration across the reported metrics rather than the best result for every individual metric.

The sensitivity analysis was conducted on an internal validation split; the official TextZoom test set and final Real-CE evaluation set were not used to select these coefficients. The results support the use of the shared default weights under the evaluated settings, but do not establish that the same coefficients are optimal for all datasets or degradation distributions.

Table 9. Sensitivity analysis of OCR and edge loss weights on the internal validation split.

Configuration	λ_ocr	λ_edge	PSNR ↑ (dB)	SSIM ↑	LPIPS ↓	Acc ↑ (%)	CER ↓
OCR-low	0.05	0.10	25.04 ± 0.05	0.882 ± 0.002	0.175 ± 0.002	82.9 ± 0.4	0.174 ± 0.004
Base	0.10	0.10	25.11 ± 0.04	0.885 ± 0.001	0.170 ± 0.002	83.7 ± 0.3	0.166 ± 0.003
OCR-high	0.20	0.10	25.07 ± 0.05	0.883 ± 0.002	0.174 ± 0.002	83.9 ± 0.4	0.164 ± 0.003
Edge-low	0.10	0.05	25.00 ± 0.05	0.880 ± 0.002	0.180 ± 0.003	83.2 ± 0.4	0.171 ± 0.004
Edge-high	0.10	0.20	25.08 ± 0.04	0.884 ± 0.001	0.172 ± 0.002	83.6 ± 0.3	0.165 ± 0.003

Values are reported as mean ± standard deviation over five runs on the internal validation split. The pixel and perceptual loss weights remain fixed at 1.00 and 0.01, respectively. OCR-low and OCR-high vary only the OCR loss weight, whereas Edge-low and Edge-high vary only the edge loss weight. The official TextZoom test set and final Real-CE evaluation set were not used for hyperparameter selection.

4.5. Independent Recognition and Language-Specific Results

To examine whether the reported recognition improvements depend on using PaddleOCR as both the training-supervision recognizer and the primary evaluation recognizer, we performed an additional evaluation using frozen EasyOCR (Table 10). EasyOCR was not used for OCR-prior generation, reconstruction training, checkpoint selection, or hyperparameter tuning. Recognition results were compared under the same evaluation references and text-normalization procedure.

On TextZoom, the exact-match accuracy improvement from RRDB to the full model was 11.0 percentage points under both PaddleOCR (72.4% to 83.4%) and EasyOCR (71.0% to 82.0%). On Real-CE, the corresponding improvements were 2.4 percentage points under PaddleOCR (47.8% to 50.2%) and 2.3 percentage points under EasyOCR (45.6% to 47.9%). Thus, the observed improvements over RRDB were largely preserved under the independent recognizer for both datasets.

The OCR-loss ablation provides an additional check. Without the OCR supervision loss, EasyOCR accuracy remained at 81.7% on TextZoom and 47.5% on Real-CE, compared with 82.0% and 47.9% for the corresponding full models. Restoring the OCR loss

therefore yielded additional improvements of 0.3 and 0.4 percentage points, respectively. These results indicate that the improvements observed with EasyOCR cannot be attributed solely to the direct OCR-loss supervision used during training.

Nevertheless, the independent-recognizer evaluation covers RRDB and selected variants of the proposed framework, rather than all external baseline methods. It therefore supports the robustness of the observed within-framework improvements to a change of evaluation recognizer, but does not independently establish the ranking of all methods in Tables 2 and 4.

Table 10. Independent-recognizer evaluation with and without OCR supervision.

Dataset	Configuration	λ_{ocr}	PaddleOCR Acc (%)	PaddleOCR CER	EasyOCR Acc (%)	EasyOCR CER
TextZoom	RRDB	—	72.4	0.238	71.0	0.292
TextZoom	Without OCR loss	0	83.0	0.169	81.7	0.190
TextZoom	Full model	0.1	83.4	0.164	82.0	0.187
Real-CE	RRDB	—	47.8	0.305	45.6	0.324
Real-CE	Without OCR loss	0	49.9	0.287	47.5	0.303
Real-CE	Full model	0.1	50.2	0.283	47.9	0.299

EasyOCR was used only for post-training independent recognition evaluation and was excluded from prior generation, training supervision, model selection, and hyperparameter tuning. The same reference transcriptions, normalization rules, and valid evaluation units were applied for PaddleOCR and EasyOCR. The parameter λ_ocr denotes the reconstruction OCR-loss coefficient: 0 for the configuration without OCR supervision and 0.1 for the full model. The table evaluates RRDB and selected variants of the proposed framework; external baseline rankings under EasyOCR are not reported.

Table 11 presents the five Real-CE configurations evaluated across language groups. G0 is RRDB; G1 adds conditional diffusion; G2 adds text, position, and structure priors with cross-attention; G3 adds Mamba with equal-weight prior fusion; the full model adds dynamic gating. Model settings, data splits, inference settings, and evaluation rules were held fixed for the language-group summaries.

Table 11. Incremental ablation across Real-CE language groups.

Configuration	Chinese PSNR (dB)	Chinese Acc (%)	English PSNR (dB)	English Acc (%)	Mixed PSNR (dB)	Mixed Acc (%)	Overall PSNR (dB)	Overall Acc (%)
G0: RRDB	23.72	45.7	24.35	54.3	23.28	37.4	23.91	47.8
G1: +diffusion	23.89	46.6	24.50	55.0	23.47	38.3	24.08	48.6
G2: +priors and CA	24.06	47.5	24.64	55.8	23.65	39.3	24.24	49.5
G3: +Mamba	24.14	48.0	24.71	56.1	23.74	39.8	24.32	49.9
Full: +dynamic gate	24.20	48.3	24.76	56.4	23.81	40.1	24.38	50.2

Chinese, English, and mixed Chinese–English groups contain 116, 101, and 44 scenes. Overall metrics are calculated from all valid aligned regions or transcribed text instances; scene counts are not aggregation weights. CA denotes cross-attention. G1 to G2 jointly adds text, position, and structure priors and cross-attention, so this stage does not isolate cross-attention or LLM correction. G3 retains equal-weight prior fusion.

From G0 to the full model, the Chinese, English, and mixed Chinese–English groups gained 0.48, 0.41, and 0.53 dB PSNR and 2.6, 2.1, and 2.7 percentage points Acc, respectively (Table 11). Each group improved after adding diffusion and then the combined prior/cross-attention stage; the later Mamba and gating stages yielded smaller gains. Mixed-language Acc remained 40.1%, below the English group's 56.4%. Because G1 to G2

added all three priors and cross-attention together, this difference measures their combined conditioning effect. It cannot isolate cross-attention or LLM correction in each language. The staged comparisons do not establish identical single-module effect sizes across languages or zero-shot transfer from one jointly trained multilingual checkpoint.

4.6. Computational Cost and Reconstruction Results

We evaluated the computational cost of the reconstruction networks on a single NVIDIA RTX 4090 under the TextZoom configuration. Parameter count, peak training GPU memory, and single-image reconstruction latency were measured separately to distinguish model complexity, training resource consumption, and online inference efficiency.

Online reconstruction latency was measured at batch size 1 using 50 warm-up runs followed by 500 timed runs. Timing included RRDB reconstruction, prior encoding, and DDIM diffusion sampling, but excluded offline OCR extraction, LLM correction, and cache generation.

Peak training GPU memory was recorded separately using a TextZoom micro-batch size of 8 and gradient accumulation over 2 steps. The reported values represent the maximum CUDA memory reserved by the PyTorch caching allocator during the measured training execution. Unlike the memory footprint of model parameters or individual feature tensors, peak reserved memory reflects the memory managed by the allocator during training, including allocations associated with intermediate activations, gradients, optimizer states, and temporary computational workspaces. Consequently, these values should not be interpreted as the GPU memory required for single-image inference.

Model parameter counts were obtained from the instantiated reconstruction networks and excluded the external PaddleOCR and Qwen2.5-7B-Instruct models. Resource comparisons refer to the complete evaluated configurations rather than isolated bottleneck operators. The corresponding results are reported in Table 12.

Table 12. Parameter counts, peak training reserved memory, and online reconstruction latency of the evaluated networks.

Model	Parameters (M)	Peak Training Reserved Memory (GB)	Latency per Image (ms)
B0: RRDB	16.70	3.8	48.00
B1: RRDB + diffusion	27.80	11.4	141.00
Full model	31.32	13.0	153.08

All measurements were conducted on an NVIDIA RTX 4090 under the TextZoom configuration. Peak training reserved memory was recorded using the PyTorch CUDA caching allocator with a micro-batch size of 8 and gradient accumulation over 2 steps. Single-image reconstruction latency was measured separately at batch size 1. Parameter counts exclude the external OCR models and LLMs, and inference timing excludes offline candidate extraction, LLM correction, and cache generation.

Adding diffusion to RRDB increased latency from 48.00 to 141.00 ms; the full model required 153.08 ms. Diffusion sampling remained the main source of additional online reconstruction latency relative to RRDB. The reported parameter counts and resource measurements characterize the complete network configurations, and differences in peak training reserved memory should not be attributed exclusively to the bottleneck sequence operators. End-to-end deployment would also incur OCR processing, LLM inference, and cache creation.

Figure 6 presents representative reconstruction results for Chinese, English, and mixed Chinese–English text. The LR input, bicubic interpolation, RRDB, PEAN, the full

model, and the HR reference are compared using identical crops and display scales. Bicubic interpolation retains substantial blur around character boundaries. RRDB improves the main outlines, although some fine strokes remain indistinct. In the selected examples, the full model produces clearer stroke separation and more continuous character contours, with character shapes more closely resembling the HR references. The comparison with PEAN provides an additional reference for examining local stroke details. These examples illustrate visual reconstruction quality, while character recognition performance is assessed through the quantitative results reported above.

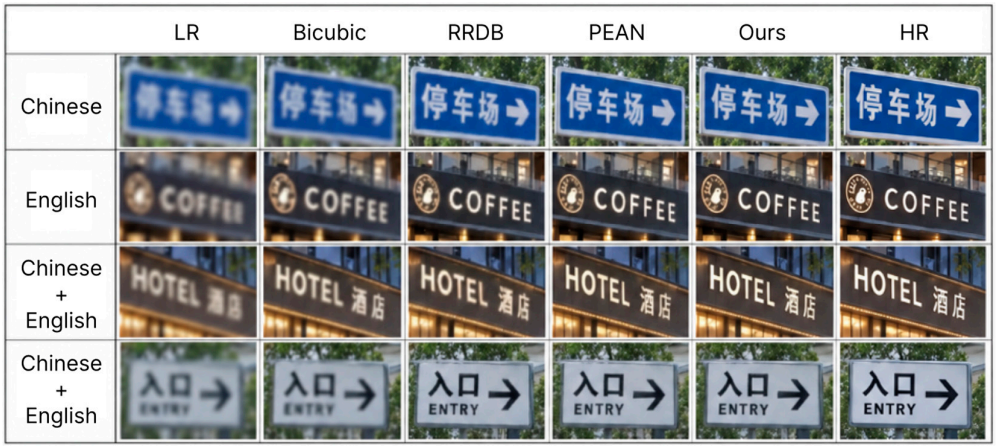


Figure 6. Qualitative comparison of Chinese, English, and mixed Chinese–English scene text. Columns from left to right show LR, bicubic interpolation, RRDB, PEAN, the proposed full model, and HR. Identical crops and display scales are used across methods.

4.7. Failure Analysis

To evaluate reconstruction performance under different degradation conditions, we further analyzed all 4373 TextZoom test samples according to their dominant visible degradation in the low-resolution (LR) input. Each sample was assigned to one of seven mutually exclusive categories: severe blur, local occlusion, noise, uneven illumination, low contrast, perspective distortion, and other routine degradation. When multiple degradations were present, the dominant category was determined using the predefined classification rule. The categorization was based exclusively on the LR observations, without consulting the reconstructed images or ground-truth transcriptions. The same category assignments were used to evaluate RRDB and the proposed full model.

It should be emphasized that the degradation-based partition differs from the standard Easy, Medium, and Hard difficulty subsets used for benchmark evaluation. The three difficulty subsets are evaluated independently and combined using the unweighted macro-average defined in Equation (16). In contrast, the seven degradation categories constitute a separate partition of the complete test set. Statistics pooled over these degradation categories therefore represent evaluation across all 4373 samples and should not be interpreted as the difficulty-subset macro-average reported in Table 2.

Within each degradation category, exact-match accuracy is calculated as the proportion of correctly recognized samples, while corpus-level CER is computed as the total number of character-level edit operations divided by the total number of reference characters. A reconstruction is considered a relative failure when its per-sample CER exceeds that of RRDB under the same recognition configuration. The corresponding failure rate is the proportion of such samples within each degradation group. Table 13 reports the recognition performance and relative failure rates under the seven degradation conditions.

Table 13. Recognition performance and relative failure rates under different dominant degradation conditions.

Dominant Degradation	Samples	RRDB Acc (%)	Full Acc (%)	RRDB CER	Full CER	Failure Rate (%)
Severe blur	406	45.1	68.5	0.360	0.310	8.6
Local occlusion	287	50.2	70.0	0.330	0.280	8.0
Noise	400	65.0	82.5	0.260	0.200	5.0
Uneven illumination	350	68.0	84.0	0.230	0.170	4.3
Low contrast	352	59.9	78.4	0.290	0.230	5.4
Perspective distortion	321	65.1	80.4	0.260	0.190	4.7
Other routine degradation	2257	86.3	90.1	0.173	0.086	0.8

The degradation-stratified results in Table 13 demonstrate that reconstruction performance varies considerably across different visual conditions. Severe blur and local occlusion remain particularly challenging, with the proposed full model achieving exact-match accuracies of 68.5% and 70.0% and CER values of 0.310 and 0.280, respectively. For noise and uneven illumination, the corresponding accuracies increase to 82.5% and 84.0%, with CER values of 0.200 and 0.170. The low-contrast and perspective-distortion groups achieve exact-match accuracies of 78.4% and 80.4%, respectively, with corresponding CER values of 0.230 and 0.190. The other-routine-degradation group achieves the highest accuracy of 90.1% and the lowest CER of 0.086.

Compared with RRDB, the proposed full model achieves higher exact-match accuracy and lower CER across all seven degradation categories. Nevertheless, the relative failure rates remain higher under severe blur (8.6%) and local occlusion (8.0%), suggesting that missing or obscured character strokes continue to limit reliable reconstruction under challenging conditions. In contrast, the other-routine-degradation group exhibits a substantially lower relative failure rate of 0.8%.

It is important to distinguish the pooled evaluation over these degradation groups from the macro-averaged benchmark results. For the proposed full model, the accuracy calculated over all 4373 samples is 83.9%, whereas the unweighted average of the Easy, Medium, and Hard subset accuracies is 83.4%. Similarly, the pooled corpus-level CER is 0.155, compared with a difficulty-subset macro-average of 0.164. These differences result from the distinct aggregation procedures and the unequal sample and reference-character distributions across the difficulty subsets. Accordingly, the pooled degradation-group statistics are not used to replace the macro-averaged benchmark metrics reported in Table 2.

Furthermore, CER and string error rate characterize different aspects of recognition performance. The string error rate measures the proportion of incorrectly recognized strings, whereas corpus-level CER is based on the total character edit distance normalized by the total number of reference characters. Therefore, the two metrics are not necessarily complementary, and CER may exceed the string error rate when incorrectly recognized strings contribute multiple character-level errors. Both metrics are calculated independently according to their respective definitions.

5. Discussion

5.1. Prior Reliability and Visual Evidence

The experimental findings should be interpreted by distinguishing the contribution of constrained OCR correction from those of the remaining reconstruction components. The correction interface determines which OCR-supported textual substitutions are admissible before prior encoding, whereas reliability-aware conditioning adjusts how the adopted text is combined with positional and structural evidence during reconstruction. These mechanisms operate at different stages and therefore require separate evaluation.

The five-seed P1-O/P1-L comparison on TextZoom shows that replacing raw OCR text with constrained-correction text improves PSNR by 0.08 dB and exact-match accuracy by 0.7 percentage points within the base text-conditioned configuration. The additional full-model experiments further demonstrate mean improvements of 0.07 dB in PSNR and 0.6 percentage points in accuracy on both TextZoom and Real-CE. The paired 95% confidence intervals exclude zero for both metrics under the evaluated conditions.

These results provide direct evidence of the downstream reconstruction contribution of constrained OCR correction in both reduced and complete model configurations. Nevertheless, the observed effects are modest, and the controlled comparisons isolate the adopted textual prior rather than the contribution of each individual correction constraint. The overall improvement of the full framework over RRDB reflects the combined effects of residual diffusion, complementary prior conditioning, cross-attention, bottleneck sequence modeling, and reliability gating. It should therefore not be attributed to linguistic correction alone.

The prior-combination ablations further clarify why the three branches are used together. Textual information supplies character-content cues, positional information associates these cues with detected text locations, and structural information preserves image-derived stroke and edge evidence. The gains from adding position or structure to the corrected-text configuration indicate that these cues are complementary within the tested base framework. However, the observed improvements depend on the specific combination, and the current ablations do not establish that each branch contributes an independent additive gain under every configuration. They support the use of complementary evidence rather than the claim that any single prior is sufficient for reliable reconstruction.

The acceptance rules restrict permitted edits without guaranteeing that the accepted string is correct. The true character may be missing from the candidate set, and an incorrect high-confidence prediction may bypass correction. The equal-length rule also prevents correction of insertion and deletion errors. Reverting to the original text retains any errors it already contains. The post-correction miscorrection rate measures changes to initially correct OCR strings, not incorrect characters synthesized in the final image. The failures under severe blur and occlusion are consistent with a broader limitation: linguistic plausibility cannot replace weak visual evidence.

Reliability gating operates at the sample level and adjusts the relative contributions of the prior branches rather than directly identifying erroneous characters. In the confidence-stratified analysis, the low-confidence group showed larger average gains from dynamic gating than the higher-confidence group. The gate also assigned a smaller mean textual weight and a larger mean structural weight to the low-confidence group. Furthermore, in the diagnostic subset of 612 samples with incorrect adopted textual priors and valid text–position–structure alignment, dynamic gating achieved higher average PSNR and exact-match accuracy than fixed global weights. This finding is limited to samples with usable complementary prior evidence and should not be generalized to all incorrect OCR transcriptions. These findings provide complementary evidence of confidence-dependent weighting and improved average reconstruction under incorrect textual conditioning. However, OCR confidence is not equivalent to transcription correctness, and the results do not establish that every incorrect prior is suppressed or that all affected samples benefit. A single set of sample-level branch weights also cannot independently accommodate different reliability levels among characters or regions within the same image. The conclusions are therefore limited to the tested model configuration and evaluation protocol.

Although no additional activation map is introduced, the behavior of the conditioning mechanism can be examined through the learned branch weights and the controlled component comparisons. The gate assigns different relative contributions to textual,

positional, and structural evidence according to the sample-level reliability vector. The confidence-stratified results show lower textual weights and higher structural weights in the low-confidence group, while the ablation results quantify the performance changes associated with alternative conditioning configurations. These observations provide indirect evidence about how the conditioning components contribute to reconstruction, but they do not directly visualize spatial attention or establish causal attribution to individual image regions. A dedicated activation-map analysis would be needed to examine such spatial effects more directly.

5.2. Symmetry and Asymmetry in Character Structure

Symmetry and asymmetry are complementary characteristics of the geometric structures involved in scene text reconstruction. Some glyphs exhibit approximate reflectional relationships between strokes or components, whereas others depend on asymmetric positions, openings, or orientations for their identity. These relationships are affected by blur, occlusion, geometric distortion, and image resampling. Inaccurate reconstruction can therefore produce two different structural errors: disruption of approximately balanced stroke arrangements and artificial regularization of distinctive asymmetric details. The relevant reconstruction objective is to preserve the geometry supported by the observation, rather than to maximize symmetry indiscriminately.

The proposed framework addresses this geometry-sensitive setting through complementary visual and textual conditioning. The LR-derived structural encoder represents gradient magnitude, orientation, and the spatial arrangement of observed strokes. Positional priors associate the encoded information with detected text regions, whereas textual priors provide candidate character identities. Reliability gating adjusts the relative contributions of these sources when OCR-derived information is uncertain. In this way, the framework can incorporate evidence about both approximately symmetric structures and character-defining asymmetric details without explicitly imposing reflection constraints. The prior-combination results in Table 6 show that introducing structural evidence improves general reconstruction and recognition performance under the evaluated base configuration. However, these improvements cannot be attributed specifically to the recovery of symmetric patterns.

Accordingly, the relationship of this study to symmetry and asymmetry concerns the preservation of character geometry during image restoration and the integration of complementary structural evidence into a recognition-oriented generative framework. The method does not estimate symmetry axes, enforce reflection equivariance, or introduce a symmetry-specific optimization objective. Moreover, the reported PSNR, SSIM, LPIPS, and recognition metrics evaluate overall reconstruction quality rather than symmetry preservation separately. The current experiments therefore do not establish a distinct symmetry-specific performance contribution. A dedicated evaluation of approximately symmetric and asymmetric glyph structures, including their corresponding HR references, would be required to quantify this aspect directly. We consider this distinction important for interpreting the present contribution without extending its claims beyond the available evidence.

5.3. Scope and Cost of Sequence Modeling

The bottleneck comparison indicates that Mamba and the lightweight Transformer achieve comparable reconstruction quality under the evaluated TextZoom configuration. The complete Mamba-based reconstruction network contains 31.32 M parameters, compared with 32.44 M for the Transformer-based configuration. Reconstruction latency decreases from 160.90 to 153.08 ms per image. These findings support a reduction in parameter count and online reconstruction latency for the tested Mamba configuration.

The GPU memory results represent peak training reserved memory, rather than single-image inference memory. Although the Mamba-based configuration recorded 13.0 GB compared with 13.8 GB for the Transformer-based configuration, these measurements reflect complete training executions and are influenced by intermediate activations, optimizer states, temporary workspaces, and CUDA allocator behavior. Therefore, the observed difference does not establish an intrinsic 0.8 GB memory saving attributable to the Mamba operator itself.

The horizontal sequence length is 16 for TextZoom training crops and 32 for Real-CE training crops. The current experiments do not establish an asymptotic computational advantage for substantially longer sequences or comparable resource savings across other hardware platforms. The fixed left-to-right scan also limits the applicability of the observations to vertical, curved, and multidirectional text.

Diffusion sampling remains the main additional source of online reconstruction latency relative to RRDB. Under the evaluated TextZoom configuration, the full reconstruction network requires 153.08 ms per image at batch size 1, while its peak training reserved memory is 13.0 GB under the separate training configuration. These measurements exclude the external OCR and LLM preprocessing stages. Consequently, the reported reconstruction-network costs do not represent the complete end-to-end deployment cost and do not establish suitability for resource-constrained devices.

5.4. Evaluation Scope and Relation to Previous Work

The evaluation protocols define the scope of the conclusions. TextZoom and Real-CE use separately trained models, different scale factors, and different evaluation units; their absolute scores are not measurements of an equivalent task. Fine-grained single-factor ablations are concentrated on TextZoom. Real-CE includes Chinese, English, and mixed-language groups, but its smaller set of 261 valid scenes and grouped comparisons do not establish identical effects for each isolated module across languages. Coverage of vertical text, curved text, artistic fonts, strong reflections, and severe occlusion is limited. The study does not test zero-shot transfer of a single multilingual checkpoint.

The degradation-stratified analysis in Section 4.7 extends the evaluation to severe blur, local occlusion, noise, uneven illumination, low contrast, and perspective distortion. However, the categories are based on the dominant visible degradation in the LR input rather than controlled manipulation of individual degradation factors. The results therefore provide observational evidence of performance under the evaluated conditions, not a causal estimate of the effect of each degradation type or proof of robustness to all real-world scenarios.

The independent EasyOCR evaluation and the OCR-loss ablation provide complementary checks on dependence on PaddleOCR. The improvement in exact-match accuracy from RRDB to the full model was 11.0 percentage points under both recognizers on TextZoom and was 2.4 and 2.3 percentage points under PaddleOCR and EasyOCR, respectively, on Real-CE. Most of the improvement also remained after the OCR supervision loss was removed. These observations suggest that the reported within-framework recognition gains are not solely a consequence of evaluating with the same recognizer used for training supervision.

However, PaddleOCR remains part of prior construction, OCR-loss supervision, and the primary evaluation protocol. A shared-recognizer effect cannot be ruled out completely. Furthermore, the EasyOCR comparison was limited to RRDB and selected configurations of the proposed method; external baseline methods were not comprehensively reevaluated using EasyOCR. Therefore, the relative rankings of all methods under an independent recognizer remain undetermined. The results should be interpreted as evidence of consistency across the two tested recognizers for the evaluated configurations,

rather than proof of recognizer-independent superiority over all external baselines. Broader cross-recognizer evaluation and matched analysis of LR versus upsampled-image OCR remain important limitations of the present study.

The present study builds upon our previous work on OCR priors and cross-scale attention [11], while addressing a different methodological focus. The earlier framework emphasized OCR-guided reconstruction and cross-scale feature interaction. In contrast, the present method introduces a constrained OCR–LLM correction interface with deterministic validation and fallback, integrates textual, positional, and structural priors through reliability-aware conditioning, and incorporates Mamba-assisted residual diffusion for reconstruction. These developments focus on the reliability and utilization of OCR-derived information rather than solely on improving conventional reconstruction metrics.

For completeness, the previously published method [11] achieved 24.85 dB PSNR, 0.877 SSIM, and 82.6% recognition accuracy on TextZoom. The proposed full model reports corresponding values of 25.06 dB, 0.883, and 83.4%, representing numerical differences of 0.21 dB, 0.006, and 0.8 percentage points, respectively. The TATT and DPMN reference results in Table 2 also coincide with those reported in [11]. This relationship is explicitly acknowledged to distinguish previously reported baseline measurements from the performance evaluation of the present framework.

These results provide a quantitative reference for understanding the progression between the two studies. Nevertheless, the numerical differences should not be interpreted as a fully controlled estimate of improvement unless identical evaluation conditions are established. The main methodological contributions of the present framework are further examined through the controlled component comparisons in Section 4, including OCR correction, prior conditioning, and bottleneck sequence modeling. Accordingly, the significance of the present work lies in its additional mechanisms for managing uncertain OCR priors and reconstructing text images under the evaluated conditions, rather than in claiming a substantial numerical advantage over the earlier framework.

The sensitivity analysis provides additional evidence about the stability of the selected loss weights under the evaluated validation setting. Using the same coefficients on TextZoom and Real-CE avoids dataset-specific loss tuning and offers a limited test of a shared training objective across two distinct datasets. However, the observed results do not establish that the default coefficients are universally optimal. Differences in language composition, image degradation, scale factor, and evaluation units may affect the preferred balance among pixel, perceptual, OCR, and edge supervision. The conclusions regarding weight generalizability are therefore restricted to the two datasets and configurations examined in this study.

The interpretability analysis in this study relies on branch-weight statistics and controlled ablation comparisons rather than spatial activation visualizations such as Grad-CAM. These analyses help characterize the relative use of the conditioning branches and their associated performance changes, but they do not reveal the precise image regions responsible for individual reconstruction decisions. Spatially resolved attribution and attention visualization remain useful directions for future work.

6. Conclusions

We presented a residual diffusion method for Chinese and English scene text image super-resolution that combines constrained OCR–LLM correction, complementary text, position, and structural priors, and bottleneck Mamba modeling. Candidate restrictions and deterministic validation make the adopted text traceable, while cross-attention and sample-level gating organize its use alongside visual evidence. Under the evaluated protocols, the complete model achieved PSNR values of 25.06 dB and 24.38 dB on TextZoom

and Real-CE, respectively, with exact-match accuracies of 83.4% and 50.2%. Controlled comparisons indicated a modest reconstruction benefit from corrected text and a favorable resource trade-off for Mamba relative to the tested lightweight Transformer. Independent-recognizer evaluation supported improved readability relative to RRDB. These results support the tested integration; candidate coverage, sample-level reliability resolution, severe degradation, limited language-specific ablations, scan direction, and full-system cost bound its applicability.

From the perspective of symmetry and asymmetry, the reconstruction task requires preserving both approximately balanced stroke arrangements and distinctive asymmetric character details. The present method incorporates relevant structural evidence without enforcing symmetry-specific constraints, while direct quantitative validation of symmetry preservation remains outside the scope of the reported experiments.

Author Contributions: Conceptualization, X.Q. and J.L.; methodology, X.Q.; software, X.Q. and P.X.; validation, X.Q., P.X. and C.L.; formal analysis, X.Q. and Y.F.; investigation, X.Q. and C.L.; data curation, X.Q., P.X. and C.L.; writing—original draft preparation, X.Q.; writing—review and editing, P.X., C.L., Y.F. and J.L.; supervision, J.L.; project administration, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: During the revision of this manuscript, the authors used OpenAI ChatGPT, based on the GPT-6 model, solely to assist with English grammar, sentence structure, and readability. All AI-assisted outputs were reviewed and edited by the authors, who take full responsibility for the content of the publication. Generative AI was not used to generate the research methodology, experimental data, results, analyses, figures, or conclusions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, W.; Xie, E.; Liu, X.; Wang, W.; Liang, D.; Shen, C.; Bai, X. Scene Text Image Super-Resolution in the Wild. In *Computer Vision—ECCV 2020*; Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2020; Volume 12355, pp. 650–666. https://doi.org/10.1007/978-3-030-58607-2_38.
2. Ma, J.; Liang, Z.; Xiang, W.; Yang, X.; Zhang, L. A Benchmark for Chinese-English Scene Text Image Super-Resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 19395–19404. https://doi.org/10.1109/ICCV51070.2023.01782.
3. Ma, J.; Liang, Z.; Zhang, L. A Text Attention Network for Spatial Deformation Robust Scene Text Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 5901–5910. https://doi.org/10.1109/CVPR52688.2022.00582.
4. Ma, J.; Guo, S.; Zhang, L. Text Prior Guided Scene Text Image Super-Resolution. *IEEE Trans. Image Process.* **2023**, *32*, 1341–1353. https://doi.org/10.1109/TIP.2023.3237002.
5. Zhu, S.; Zhao, Z.; Fang, P.; Xue, H. Improving Scene Text Image Super-Resolution via Dual Prior Modulation Network. *Proc. AAAI Conf. Artif. Intell.* **2023**, *37*, 3843–3851. https://doi.org/10.1609/aaai.v37i3.25497.
6. Noguchi, C.; Fukuda, S.; Yamanaka, M. Scene Text Image Super-Resolution Based on Text-Conditional Diffusion Models. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 4–8 January 2024; pp. 1474–1484. https://doi.org/10.1109/WACV57701.2024.00151.
7. Zhao, Z.; Xue, H.; Fang, P.; Zhu, S. PEAN: A Diffusion-Based Prior-Enhanced Attention Network for Scene Text Image Super-Resolution. In Proceedings of the 32nd ACM International Conference on Multimedia, Melbourne, VIC, Australia, 28 October–1 November 2024; pp. 9769–9778. https://doi.org/10.1145/3664647.3680974.

8. Boros, E.; Ehrmann, M.; Romanello, M.; Najem-Meyer, S.; Kaplan, F. Post-Correction of Historical Text Transcripts with Large Language Models: An Exploratory Study. In Proceedings of the 8th Joint SIGHUM Workshop on Computational Linguistics for Cultural Heritage, Social Sciences, Humanities and Literature (LaTeCH-CLfL 2024), St. Julians, Malta, 22 March 2024; pp. 133–159. https://doi.org/10.18653/v1/2024.latechclfl-1.14.
9. Gu, A.; Dao, T. Mamba: Linear-Time Sequence Modeling with Selective State Spaces. In Proceedings of the First Conference on Language Modeling, Philadelphia, PA, USA, 7–9 October 2024. Available online: https://openreview.net/forum?id=tEYskw1VY2 (accessed on 28 September 2026).
10. Guo, H.; Li, J.; Dai, T.; Ouyang, Z.; Ren, X.; Xia, S.-T. MambaIR: A Simple Baseline for Image Restoration with State-Space Model. In *Computer Vision—ECCV 2024*; Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2025; Volume 15076, pp. 222–241. https://doi.org/10.1007/978-3-031-72649-1_13.
11. Qiu, X.; Liu, J.; Fang, C. Text Image Super-Resolution via Fusion of OCR Priors and Cross-Scale Attention. *Symmetry* **2026**, *18*, 1221. https://doi.org/10.3390/sym18071221.
12. Ye, K.; Garcia Dorado, I.; Raptis, M.; Delbracio, M.; Zhu, I.; Milanfar, P.; Talebi, H. TextSR: Diffusion Super-Resolution with Multilingual OCR Guidance. *arXiv* **2025**, arXiv:2505.23119. https://doi.org/10.48550/arXiv.2505.23119.
13. Xu, Z.; Liu, X.; Chen, Z.; Zhang, Y.; Yang, X. PRISM: Prior Rectification and Uncertainty-Aware Structure Modeling for Diffusion-Based Text Image Super-Resolution. *arXiv* **2026**, arXiv:2605.13027. https://doi.org/10.48550/arXiv.2605.13027.
14. Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; Wang, X. Vision Mamba: Efficient Visual Representation Learning with Bidirectional State Space Model. In Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria, 21–27 July 2024; Volume 235, pp. 62429–62442. Available online: https://proceedings.mlr.press/v235/zhu24f.html (accessed on 28 September 2026).
15. Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; Jiao, J.; Liu, Y. VMamba: Visual State Space Model. *Adv. Neural Inf. Process. Syst.* **2024**, *37*, 103031–103063. https://doi.org/10.52202/079017-3273.
16. Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In *Computer Vision—ECCV 2018 Workshops*; Leal-Taixé, L., Roth, S., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2019; Volume 11133, pp. 63–79. https://doi.org/10.1007/978-3-030-11021-5_5.
17. Qwen Team. Qwen2.5-7B-Instruct Model Card. Available online: https://huggingface.co/Qwen/Qwen2.5-7B-Instruct (accessed on 28 September 2026).
18. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30, Proceedings of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017*; Curran Associates Inc.: Red Hook, NY, USA, 2017. Available online: https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html (accessed on 28 September 2026).
19. Ho, J.; Jain, A.; Abbeel, P. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems 33, Proceedings of the 34th Conference on Neural Information Processing Systems, Online, 6–12 December 2020*; Curran Associates Inc.: Red Hook, NY, USA, 2020; pp. 6840–6851. Available online: https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html (accessed on 28 September 2026).
20. Song, J.; Meng, C.; Ermon, S. Denoising Diffusion Implicit Models. In Proceedings of the 9th International Conference on Learning Representations, Online, 3–7 May 2021. Available online: https://openreview.net/forum?id=St1giarCHLP (accessed on 28 September 2026).
21. Graves, A.; Fernández, S.; Gomez, F.; Schmidhuber, J. Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 369–376. https://doi.org/10.1145/1143844.1143891.
22. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image Quality Assessment: From Error Visibility to Structural Similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. https://doi.org/10.1109/TIP.2003.819861.
23. Zhang, R.; Isola, P.; Efros, A.A.; Shechtman, E.; Wang, O. The Unreasonable Effectiveness of Deep Features as a Perceptual Metric. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 586–595. https://doi.org/10.1109/CVPR.2018.00068.

24. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 295–307. https://doi.org/10.1109/TPAMI.2015.2439281.
25. Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Online, 11–17 October 2021; pp. 1833–1844. https://doi.org/10.1109/ICCVW54120.2021.00210.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.