DocRevive: A Unified Pipeline for Document Text Restoration

Supplementary Material

9. Dataset Construction Details

This supplementary section provides the full construction details of the Occluded Pages Restoration Benchmark (OPRB). In the current generator, we choose N unique source pages per class-level.

We introduce a novel benchmark dataset called Occluded Pages Restoration Benchmark (OPRB) designed to evaluate document restoration under a variety of real-world like simulated occlusions. The dataset consists of degradation into six occlusion classes, each with distinct visual and semantic properties. Four *standard* classes, Black Ink, Burnt, Whitener, Dust, target a quantitatively controlled pixel coverage level $T\%$ of the total document area A_D, with the current generator using three target levels $T \in \{0.5, 1.0, 1.5\}$. Opaque classes (Black Ink, Burnt, Whitener) use fully opaque patches, whereas Dust uses semi-transparent patches at opacity 0.65, through which underlying text remains partially readable. Two additional *simulation* classes, Scribble and Stamp, use the placement logic and do not target a numeric coverage percentage. Source images are partitioned such that no image appears in more than one class-level combination, ensuring all evaluation splits are strictly non-overlapping.

For standard classes, between $n \sim \mathcal{U}\{3, 7\}$ patches are selected and placed sequentially to approach the target covered area $A_T = \frac{T}{100} A_D$. Coverage is tracked pixel-precisely via a binary coverage mask M_{cov} of the same resolution as the document, and a parallel occlusion mask M_{occ} used exclusively for ground-truth refinement, which is populated only for opaque patches. For each patch P with effective non-transparent pixel area A_P, the placement scale during the initial pass is computed as

$$s = \sqrt{\frac{A_R}{A_P}}, \tag{2}$$

where $A_R = \frac{A_T - A_O}{k_{\text{left}}}$ is the per-patch share of the remaining uncovered area, k_{left} is the number of initial-pass patches left to place, and $A_O = |M_{\text{cov}}|$ denotes the cumulative covered pixel count at the time of placement. The scale is clamped to $[0.005, 10.0]$ to prevent degenerate patches. After scaling, each initial-pass patch is rotated by $\theta \sim \mathcal{U}[0°, 360°]$ and placed at a position drawn uniformly from $\mathcal{U}\big[-\frac{w_P}{4}, W - \frac{3w_P}{4}\big] \times \mathcal{U}\big[-\frac{h_P}{4}, H - \frac{3h_P}{4}\big]$, permitting partial out-of-frame placement to simulate edge occlusions. If the target is not reached within the initial n patches, up to six top-up passes add further patches scaled to the full residual area $A_T - A_O$ until $A_O \geq 0.95\, A_T$ these top-up patches are placed fully within the document bounds.

The simulation class simulates crossing and scribbling out words by placing handwritten scribble patches over random word bounding boxes. Between 5 and 6 words per image are targeted, sampled uniformly as $n \sim \mathcal{U}\{5, 6\}$, subject to validity filters, the word must belong to the `paragraph` class of the source ground truth, have length greater than two characters, contain at least one alphabetic character, not consist solely of symbols or digits, have a minimum bounding box width of 15 px, and not lie within a figure or equation exclusion zone. Each selected scribble is resized to match the word bounding box with a small random oversize factor in width $\mathcal{U}[1.00, 1.15]$ and height $\mathcal{U}[1.00, 1.20]$, placed with slight positional jitter, and tilted by $\pm 10°$ for a natural handwritten appearance. Scribbled words are entirely removed from the ground truth. Whereas, the Stamp class places exactly one semi-transparent stamp (opacity 0.60) per document in one of two modes chosen at random (a) *centre* stamp scaled to $\mathcal{U}[20, 35]\%$ of the document height and centred on the page, or (b) *corner* stamp scaled to $\mathcal{U}[8, 12]\%$ of the document height placed at the bottom-left or bottom-right corner with a 3% margin, both tilted by $\pm 15°$. Because stamps are semi-transparent, the ground truth is left entirely unmodified.

Word-level bounding boxes are refined against M_{occ} via the column-wise procedure described in Algorithm 1. For each surviving word, a column occupancy vector $\mathbf{c} \in \mathbb{R}^w$ is computed over the bbox region, where $c_j = \frac{1}{h} \sum_i M_{\text{occ}}[y_1{:}y_2,\, x_1 + j]$. The valid horizontal extent is found by scanning inward from both sides to locate columns with occupancy below a threshold $\tau_{\text{col}} = 0.30$, and trimming any interior dense wall where $c_j \geq \tau_{\text{dense}} = 0.90$. A word is removed if no valid column extent exists, if the global occlusion ratio of the trimmed region exceeds $\tau_{\text{global}} = 0.50$, or if the trimmed bbox falls below minimum dimensions ($w < 10$ px, $h < 10$ px, or area < 150 px^2). This refinement applies only to standard classes and the simulatinf scribbles removes the scribbled words entirely, and Stamp and Dust leaves all annotations intact.

After occlusion and ground-truth refinement, all images across all classes are subjected to a random global rotation $\varphi \sim \mathcal{U}[-5.0°, +5.0°]$ to simulate realistic scan misalignment, using full-canvas expansion with white fill so that no content is clipped. The rotation matrix

$$\mathbf{M} = \begin{bmatrix} \cos\varphi & -\sin\varphi & t_x \\ \sin\varphi & \cos\varphi & t_y \end{bmatrix}, \tag{3}$$

with translation offsets $t_x = \frac{W'}{2} - \frac{W}{2}$ and $t_y = \frac{H'}{2} - \frac{H}{2}$ mapping to the expanded canvas of size $W' \times H'$, is applied jointly to the image and all surviving word bounding boxes.

Algorithm 1 Column-wise Ground-Truth Refinement (Standard Mode)

Input: Word bboxes $\{B_i\} = \{(x_1, y_1, x_2, y_2)\}_i$; occlusion mask M_{occ}
Constants: $\tau_{\text{col}} = 0.30$, $\tau_{\text{dense}} = 0.90$, $\tau_{\text{global}} = 0.50$, $w_{\min} = 10$, $h_{\min} = 10$, $A_{\min} = 150$
for each $B_i = (x_1, y_1, x_2, y_2)$ **do**
  sub $\leftarrow M_{\text{occ}}[y_1{:}y_2,\ x_1{:}x_2]$; $bw \leftarrow x_2 - x_1$; $bh \leftarrow y_2 - y_1$
  if $|\text{sub}|/(bw \cdot bh) < 0.05$ **then**
    Keep B_i unchanged
    continue
  end if
  Compute column occupancy $c_j \leftarrow \frac{1}{bh}\sum_i \text{sub}[i, j]$ for $j = 0, \ldots, bw - 1$
  $l \leftarrow \min\{j : c_j < \tau_{\text{col}}\}$; $r \leftarrow \max\{j : c_j < \tau_{\text{col}}\}$
  if no valid l or r, or $r < l$ **then**
    Remove B_i
    continue
  end if
  for $j = l$ to r **do**
    if $c_j \geq \tau_{\text{dense}}$ **then**
      if $j = l$ **then**
        Remove B_i
        break
      else
        $r \leftarrow j - 1$
        break
      end if
    end if
  end for
  $fw \leftarrow r - l + 1$
  if $|\text{sub}[:, l{:}r{+}1]|/(fw \cdot bh) > \tau_{\text{global}}$ **then**
    Remove B_i
    continue
  end if
  if $fw < w_{\min}$ or $bh < h_{\min}$ or $fw \cdot bh < A_{\min}$ **then**
    Remove B_i
    continue
  end if
  Update $B_i \leftarrow (x_1 + l,\ y_1,\ x_1 + r + 1,\ y_2)$
end for
Output: Refined bounding boxes $\{B_i\}$

Each refined bbox is first expressed as a four-corner polygon $C_i = (x_1, y_1,\ x_2, y_1,\ x_2, y_2,\ x_1, y_2)$, and the corners are transformed under $\mathbf{M}$ to get the final 8-point polygon annotation format used for oriented text detection evaluation:

$$\hat{C}_i = \begin{pmatrix} \mathbf{M}[\mathbf{p}_1^\top; 1]^\top,\ \mathbf{M}[\mathbf{p}_2^\top; 1]^\top, \\ \mathbf{M}[\mathbf{p}_3^\top; 1]^\top,\ \mathbf{M}[\mathbf{p}_4^\top; 1]^\top \end{pmatrix} \tag{4}$$

All annotations are stored in the format `word` $x_0\, y_0\ \ x_1\, y_1\ \ x_2\, y_2\ \ x_3\, y_3$ `font class`, compatible with standard oriented document-text detection/recognition, layout and other benchmarks.

10. Method Details

10.1. Occlusion Detection and Blank Region Extraction

Occlusion patches are first localized using a fine-tuned YOLOv9c detector [41] trained on the OPRB dataset. The benchmark contains six degradation classes, *Black Ink*, *Burnt*, *Whitener*, *Dust*, *Scribble*, and *Stamp*. Opaque classes (*Black Ink*, *Burnt*, *Whitener*) fully obscure the underlying text, transparent classes (*Dust*, *Stamp*) allow partial visibility and no bounding boxes are omitted or trimmed, and the *Scribble* class omits out individual words.

Once the YOLO detected patches $P = \{p_1, p_2, \ldots, p_k\}$ are detected (each p_j given by absolute pixel coordinates $(x_1^j, y_1^j, x_2^j, y_2^j)$), blank regions are identified by a geometry-driven intersection method. For non-scribble patches, blanks are formed when a patch overlaps an inter-word gap, or when it forms a valid start-of-line or end-of-line blank candidate within the document text margins. Start/end blanks are accepted only after a three-sided enclosure check against neighbouring lines, duplicate emissions are suppressed, and every blank box is clamped to remain strictly within the detected occlusion patch. For scribble patches, words whose horizontal overlap with the patch exceeds 30% of the word width are grouped into a contiguous run and converted into a single anchor-bounded blank.

We tokenize a recognized text line into three parts if and only if a blank is detected: (i) the *pre-text*, (ii) the *blank*, and (iii) the *post-text*. Let $G = \{w_0, w_1, \ldots, w_n\}$ be a sorted group of word tokens for each line, where each token w_j has an associated text field $\text{text}(w_j)$. For a mid-line gap index i, we define

$$\text{PreText}_i \ = \ \text{text}(w_0) \oplus \text{text}(w_1) \oplus \cdots \oplus \text{text}(w_i),$$

$$\text{PostText}_i \ = \ \text{text}(w_{i+1}) \oplus \text{text}(w_{i+2}) \oplus \cdots \oplus \text{text}(w_n),$$

where $\oplus$ denotes string concatenation with spaces. For start-of-line blanks, PreText is empty and PostText is the full line and for end-of-line blanks, PreText is the full line and PostText is empty. Algorithm 2 constructs the formatted blank token written for each emitted gap. In RoBERTa mode, the same gap metadata are converted into per-blank masked queries for the language model.

10.2. Unified Restoration Details

Our unified document restoration framework reconstructs degraded or occluded document images by first extracting occluded regions and then restoring them using a diffusion-based text editing model. In our implementation, the overall

Algorithm 2 Prompt Token Generation for Emitted Blanks

Require: A sorted line $G = \{w_0, w_1, \ldots, w_n\}$, blank type $t \in \{\text{start}, \text{mid}, \text{end}\}$, gap index i, blank width b_w, chars-per-pixel ratio ρ, blank id α

Ensure: One formatted blank token and its metadata, or skip

$M \leftarrow \max(1, \lfloor 1.2\rho b_w \rfloor)$
if $M < 2$ **then**
  return skip
end if
if $t = \text{start}$ **then**
  $\text{PreText}_\alpha \leftarrow$ `""`
  $\text{PostText}_\alpha \leftarrow \text{text}(w_0) \oplus \cdots \oplus \text{text}(w_n)$
else if $t = \text{mid}$ **then**
  $\text{PreText}_\alpha \leftarrow \text{text}(w_0) \oplus \cdots \oplus \text{text}(w_i)$
  $\text{PostText}_\alpha \leftarrow \text{text}(w_{i+1}) \oplus \cdots \oplus \text{text}(w_n)$
else
  $\text{PreText}_\alpha \leftarrow \text{text}(w_0) \oplus \cdots \oplus \text{text}(w_n)$
  $\text{PostText}_\alpha \leftarrow$ `""`
end if
Construct Token_α as

$$\text{Token}_\alpha = \texttt{[K=}M_\alpha\texttt{]} \;\; \text{PreText}_\alpha \; \texttt{<mask>} \; \text{PostText}_\alpha$$

return Token_α and the corresponding blank metadata

process consists of several stages: (i) OCR and line grouping, (ii) YOLO-based occlusion detection, (iii) geometry-driven blank extraction when reconstruction is required, (iv) missing-text prediction, (v) adjacent-word inpainting and diffusion-based text editing, and (vi) patch compositing followed by final occlusion-region cleanup.

First, our pipeline performs OCR on the input document image I to detect text regions and generate a set of bounding boxes Z_{bbox}. The recognized text strings S are then used to group the detected boxes into lines. For each line, blank candidates are extracted by intersecting YOLO patches with OCR word gaps, including start-of-line and end-of-line cases when a three-sided enclosure test is satisfied.

After the gaps are detected, our method creates tokenized representations of each gap by concatenating the recognized words before and after the gap. For semi-transparent occlusions (*Dust* or *Stamp*), the pipeline first checks whether sufficient OCR text remains visible inside the detected patches. If the text is sufficiently visible, tokenization and missing-text prediction are skipped and the method follows an inpaint-only branch over the intersecting word boxes. Otherwise, the newly formed gaps are sent to the fine-tuned RoBERTa model, which predicts the missing strings from context. The predicted target text dictionary D is subsequently used in the text editing module, where a style patch I_s is extracted from the original document. The text editing model then generates an edited patch I_e conditioned on I_s and D_t.

To ensure consistency between the restored document and the original, a postprocessing step is performed after text editing. First, a text inpainting module [55] is applied on the adjacent word bounding boxes of the detected blank to de-occlude the visible text and cleanup minor blobs and remains of the occluding patches. The edited patches are then cleaned and integrated back into the document, after which a final occlusion-region cleaning step removes residual borders, whitespace, and inter-line artifacts. The final restored document I_{final} is obtained by placing the edited patch I_e over the original image I while also cleaning the gaps between nearby lines for visually seamless compositing.

Figure 7 presents qualitative results of DocRevive across different occlusion types. These examples show that our method can accurately localize occluded regions, recover the missing text, and restore the document while preserving the original layout, spacing, typographic structure, and overall page appearance. Unlike generic image editing approaches, the restoration remains constrained to the damaged regions, which helps maintain both structural coherence and semantic integrity. These results suggest that our method is highly optimal and an effective solution for this challenging document restoration setting.

Figure 8 compares our task with state-of-the-art industry VLMs on the same occluded documents. ChatGPT shows substantial hallucinated content and often generates random text or repeated text with complete jargon within the document body, which severely harms document integrity. Nano Banana produces fewer hallucinations than ChatGPT, but it frequently alters the global document aesthetics, it often introduces a yellowish tint, edits irrelevant regions, and in several cases blurs the missing text or replaces it with a plain white blank. In contrast, our method preserves the original document style while restoring only the intended occluded content, thereby maintaining both structural consistency and semantic integrity. Since this is still a new problem setting, further research can continue improving restoration quality and robustness.

10.3. Detailed UCSM Formulation

We define UCSM as *Unified Context Similarity Metric* for a predicted string P and the corresponding ground-truth string GT. The formulation explicitly combines intrinsic similarity between P and GT with contextual difficulty derived from the surrounding text.

We denote the Levenshtein distance between GT and P by $d_{\text{lev}}(GT, P)$, and let $|GT|$ and $|P|$ denote the lengths (in characters) of the two strings. The normalized edit similar-

Figure 7. Visualization of the proposed framework on a few OPRB document images for different types of occlusions. [Zoom in for better visualization]

ity is

$$S_{\text{edit}}(P, GT) = 1 - \frac{d_{\text{lev}}(GT, P)}{\max(|GT|, |P|)}. \quad (5)$$

This score is 1 when P exactly matches GT, and it decreases as more edits are required to transform P into GT.

To capture semantic similarity, we use cosine similarity over pretrained text embeddings. Let $\mathbf{e}(x)$ denote the embedding of a string x. Since cosine similarity lies in $[-1, 1]$,

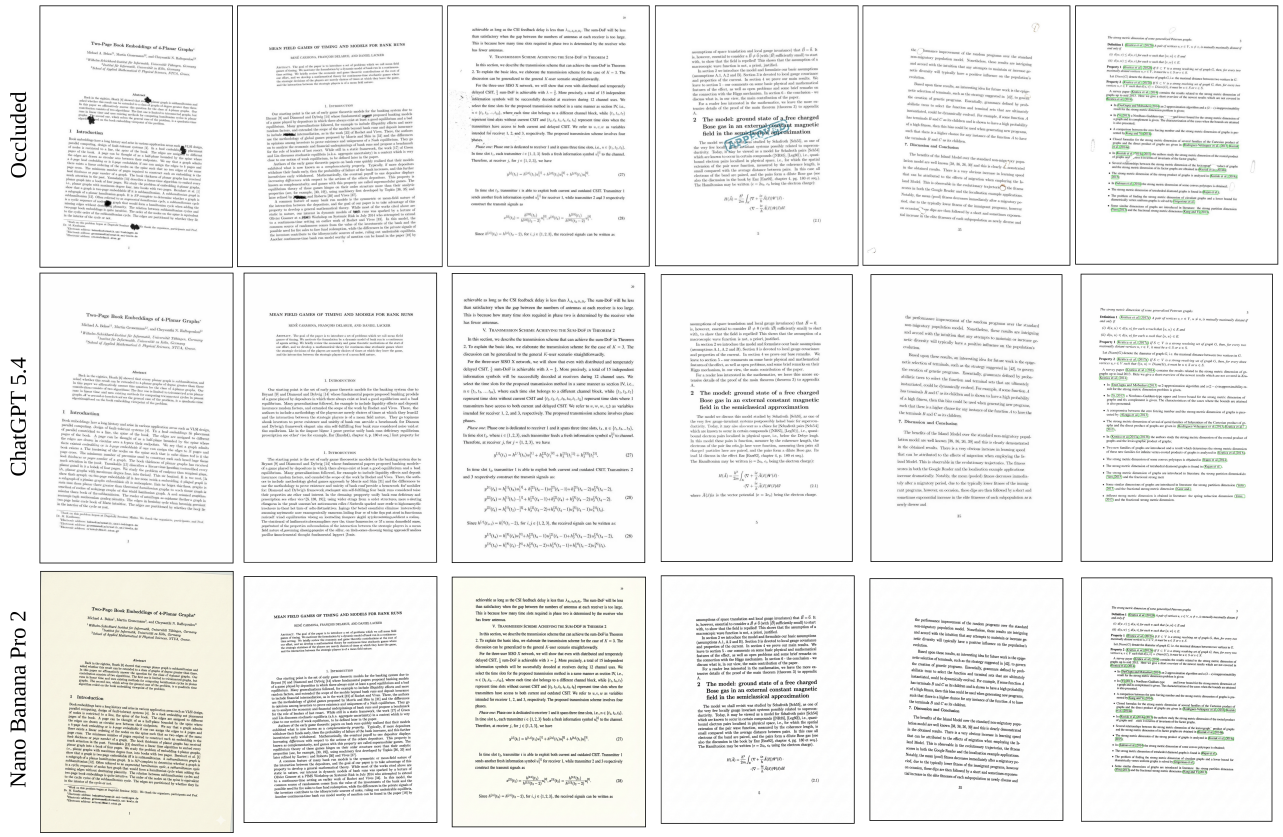

Figure 8. Visualization of the state of the art industry models on a few OPRB document images for different types of occlusions. [Zoom in for better visualization]

we rescale it to $[0, 1]$ as

$$S_{\text{sem}}(P, GT) = \frac{1}{2}\Big(\text{cosine}(\mathbf{e}(GT), \mathbf{e}(P)) + 1\Big). \tag{6}$$

We also define a length-consistency term

$$S_{\text{len}}(P, GT) = \frac{\min(|GT|, |P|)}{\max(|GT|, |P|)}. \tag{7}$$

This term equals 1 only when the two strings have identical length.

The three similarity terms are combined through a weighted geometric mean:

$$S(P, GT) = \Big[S_{\text{edit}}(P, GT)^{\alpha} \cdot S_{\text{sem}}(P, GT)^{\beta} \cdot S_{\text{len}}(P, GT)^{\gamma}\Big]^{\frac{1}{\alpha+\beta+\gamma}} \tag{8}$$

where α, β, and γ are nonnegative weights assigned to edit, semantic, and length similarity respectively. In our experiments we set $\alpha = \beta = \gamma = 1$. The geometric mean is useful here because it suppresses the combined score whenever one of the components fails badly, which is desirable for restoration quality assessment.

To incorporate contextual difficulty, suppose the preceding text is denoted by pre and the succeeding text by post. Let $\log P(\text{pre} + GT + \text{post})$ denote the log probability of the full text including GT, and let $\log P(\text{pre} + \text{post})$ denote the log probability of the context without GT. We approximate the contextual probability of the ground truth by $\log(P(GT \mid \text{pre}, \text{post}))$. After introducing calibration constants m and M, the normalized context error is

$$E_{\text{context}} = \frac{-\log P(GT \mid \text{pre}, \text{post}) - m}{M - m}, \tag{9}$$

with E_{context} clipped to the interval $[0, 1]$. A lower value of E_{context} indicates that the missing span is highly predictable from context, whereas a value near 1 indicates low predictability.

Finally, UCSM combines intrinsic similarity and contextual difficulty as

$$UCSM = S(P, GT)^{(1-E_{\text{context}})}. \tag{10}$$

If P exactly equals GT, then $S(P, GT) = 1$ and UCSM is 1 regardless of context. For imperfect predictions, deviations are penalized more strongly when the context makes

the correct answer highly predictable, which is precisely the behaviour desired for missing-text restoration.

When no embedding model is used, S_{sem} defaults to 0.5, when no LM log-probability function is used, E_{context} defaults to 0.5. When the LM term is enabled, the default calibration uses $m = -2$ and $M = 10$.

We perform a simple experiment to demonstrate how effective UCSM actually is. Table 7 demonstrates three blind spots of Edit Distance (ED) that UCSM addresses through its multi-component design. In **Example 1**, we fix the context *"We evaluate the ___ on three benchmark datasets."* and ground truth *"proposed method"*, then compare four predictions. ED ranks the valid synonym *"suggested approach"* (ED = 13) nearly as poorly as the hallucination *"random variables"* (ED = 15), whereas UCSM almost doubles the synonym's score (0.58 vs. 0.31) because S_{sem} captures that the synonym preserves meaning. In **Example 2**, both predictions have exactly ED = 5, yet one corrupts all 5 characters of a short word while the other introduces only 5 scattered OCR errors in a 44-character phrase. UCSM correctly assigns 0.00 to the total corruption and 0.87 to the minor error one, since S_{edit} normalises by string length. In **Example 3**, the same wrong prediction *"measurement"* for GT *"temperature"* gives ED = 9 in both cases, and S_{edit}, S_{sem}, and S_{len} are identical. However, when the surrounding text is highly predictive (*"... reached 300 K"*), $E_{\text{context}} \approx 0$ and UCSM penalises harshly (0.48), when the context is ambiguous (*"A ___ was taken every hour"*), $E_{\text{context}} = 0.66$ and UCSM is more lenient (0.78). ED is entirely blind to this contextual distinction. The same is also presented in Figure 9

11. Misceleneous Experiments

11.1. Comparison with Prior Document Restoration Methods

We compare DocRevive against three prior methods on a subset of 498 images from OPRB (83 per occlusion type) DocDiff [45], GSDM (standalone), our pipeline's inpainting module run in isolation without any text prediction or editing and NAFNet [10], a strong general image restoration baseline adapted to document inputs. All methods receive the same degraded images. DocRevive however uses its full pipeline. Metrics are SSIM, PSNR, and FID against clean ground-truth pages. However, our pipeline cannot be evaluated against other document based or real based dataset as because no similar fully text occluded document dataset exists to the best of our knowledge. Therefore, only the result of different methods on our dataset is presented.

Table 8 shows overall performance. DocRevive achieves the highest SSIM and PSNR, and dramatically outperforms all baselines on FID (9.00 vs. 43.25 for the next-best method). The FID gap is particularly telling that image-level similarity metrics such as SSIM and PSNR measure pixel correspondence, but FID captures perceptual distribution quality, where DocRevive's text-aware restoration produces outputs that are statistically far closer to real clean documents than any purely visual method.

Per-occlusion-type breakdown. Tables 9 report per-type SSIM and PSNR. DocRevive consistently leads across all occlusion types in SSIM and FID. The GSDM standalone baseline performs comparably to DocRevive on opaque patch types (Black Ink SSIM 0.6814 vs. 0.6816), confirming that pure inpainting suffices for opaque occlusions; the gap widens substantially on transparent classes (Dust, Stamp) and visually complex classes (Burnt, Whitener), where text-aware reconstruction is necessary. On PSNR, NAFNet is competitive on Dust and Whitener due to its effective denoising capability, yet DocRevive's semantic reconstruction gives substantially better perceptual quality as reflected by FID.

Fig. 10 presents a qualitative comparison of DocRevive against NAFNet, DocDiff, and GSDM across six distinct occlusion types. NAFNet, being a general-purpose image restoration network, struggles to handle structured degradations such as ink strokes and stamps, often leaving visible residuals or introducing blurring artifacts in the restored regions. DocDiff, while producing smoother outputs, fails to recover fine typographic details particularly under heavy occlusions like burnt and whitener damage due to its limited conditioning on document structure. GSDM shows improved perceptual quality in certain cases but exhibits hallucinated content in regions with large missing areas, such as scribble and dust degradations. In contrast, DocRevive consistently produces restorations that are both visually coherent and faithful to the ground truth across all occlusion categories, preserving text legibility, line structure, and background consistency. This robustness across diverse degradation types demonstrates the strength of our dataset-driven approach and the model's ability to disentangle occlusion from underlying document content.

Table 7. Three blind spots of Edit Distance that UCSM addresses. S_{edit}, S_{sem}, and S_{len} are defined in Eqs. (5)–(7), E_{context} in Eq. (9), UCSM in Eq. (10). Semantic embeddings are computed with all-MiniLM-L6-v2, context predictability with GPT-2.

Prediction	**ED**	S_{edit}	S_{sem}	S_{len}	E_{ctx}	**UCSM**
Example 1: Semantic Blindness Context: *"We evaluate the _____ on three benchmark datasets."* GT: *"proposed method"*						
proposed method (exact match)	0	1.000	1.000	1.000	0.000	**1.000**
proposed methoc (OCR typo)	1	0.933	0.665	1.000	0.000	0.853
suggested approach (synonym)	13	0.278	0.859	0.833	0.000	0.584
random variables (hallucination)	15	0.062	0.523	0.938	0.000	0.313
Example 2: Length Blindness Both predictions have ED = 5, but severity is opposite						
plant (GT: "where", 5 chars)	5	0.000	0.649	1.000	0.500	**0.000**
5 OCR errors in 44-char phrase	5	0.886	0.503	1.000	0.500	**0.874**
Example 3: Context Blindness GT: *"temperature"* → Pred: *"measurement"* (ED = 9 in both cases)						
Predictable: *"The _____ of the system reached 300 K."*	9	0.182	0.619	1.000	0.000	**0.483**
Ambiguous: *"A _____ was taken every hour."*	9	0.182	0.619	1.000	0.657	**0.779**

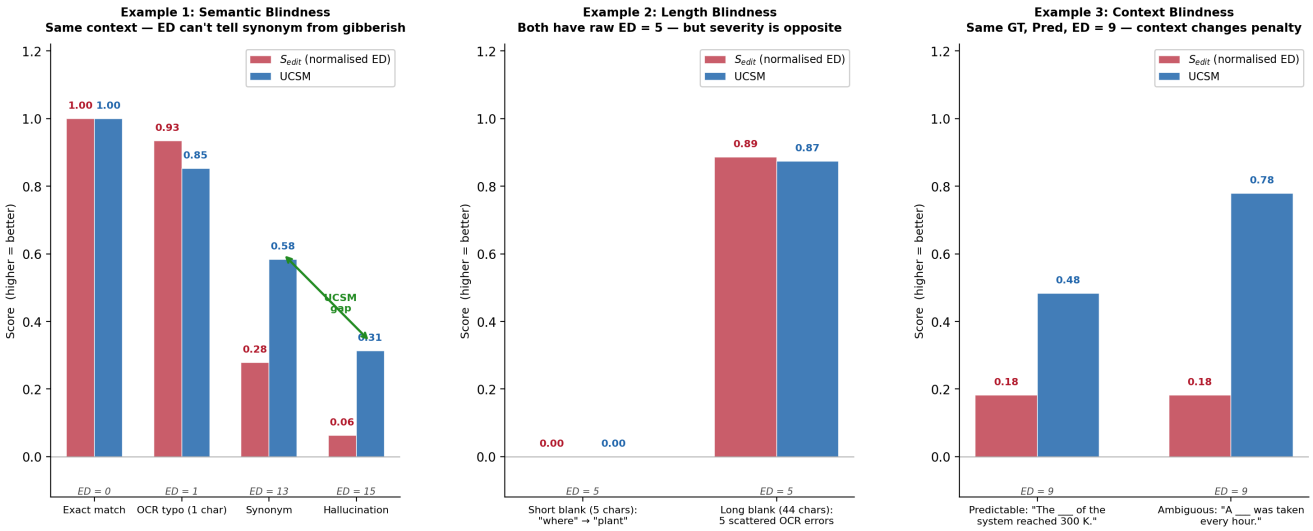

Figure 9. Graphical visualization of effectiveness of UCSM over Edit Distance [Zoom in for better visualization]

Table 8. Overall comparison on the OPRB subset. Best results in **bold**. Second best in underline.

Method	**SSIM ↑**	**PSNR ↑**	**FID ↓**
DocDiff	0.6620	16.46	43.25
GSDM	0.6634	16.58	53.78
NAFNet	0.6663	**17.42**	67.57
DocRevive (Ours)	**0.6814**	17.23	**9.00**

Table 9. Per-occlusion-type SSIM ↑ and PSNR ↑ on the OPRB. Best in **bold**. Second best in underline

	SSIM ↑						PSNR ↑					
Method	**Blk. Ink**	**Scrib.**	**Dust**	**Stamp**	**Burnt**	**Whit.**	**Blk. Ink**	**Scrib.**	**Dust**	**Stamp**	**Burnt**	**Whit.**
DocDiff	0.6792	0.6726	0.6416	0.6538	0.6748	0.6500	15.90	16.90	16.45	16.12	16.91	16.51
GSDM	0.6814	0.**6751**	0.6458	0.6556	0.6742	0.6486	16.93	17.37	15.61	15.75	17.11	16.72
NAFNet	0.6790	0.6699	0.6490	0.6609	0.6805	0.6587	16.73	**17.77**	**17.44**	**17.10**	**17.88**	**17.59**
DocRevive	**0.6816**	0.6745	**0.6833**	**0.6768**	**0.6866**	**0.6855**	**17.24**	17.09	17.26	16.83	17.48	17.50

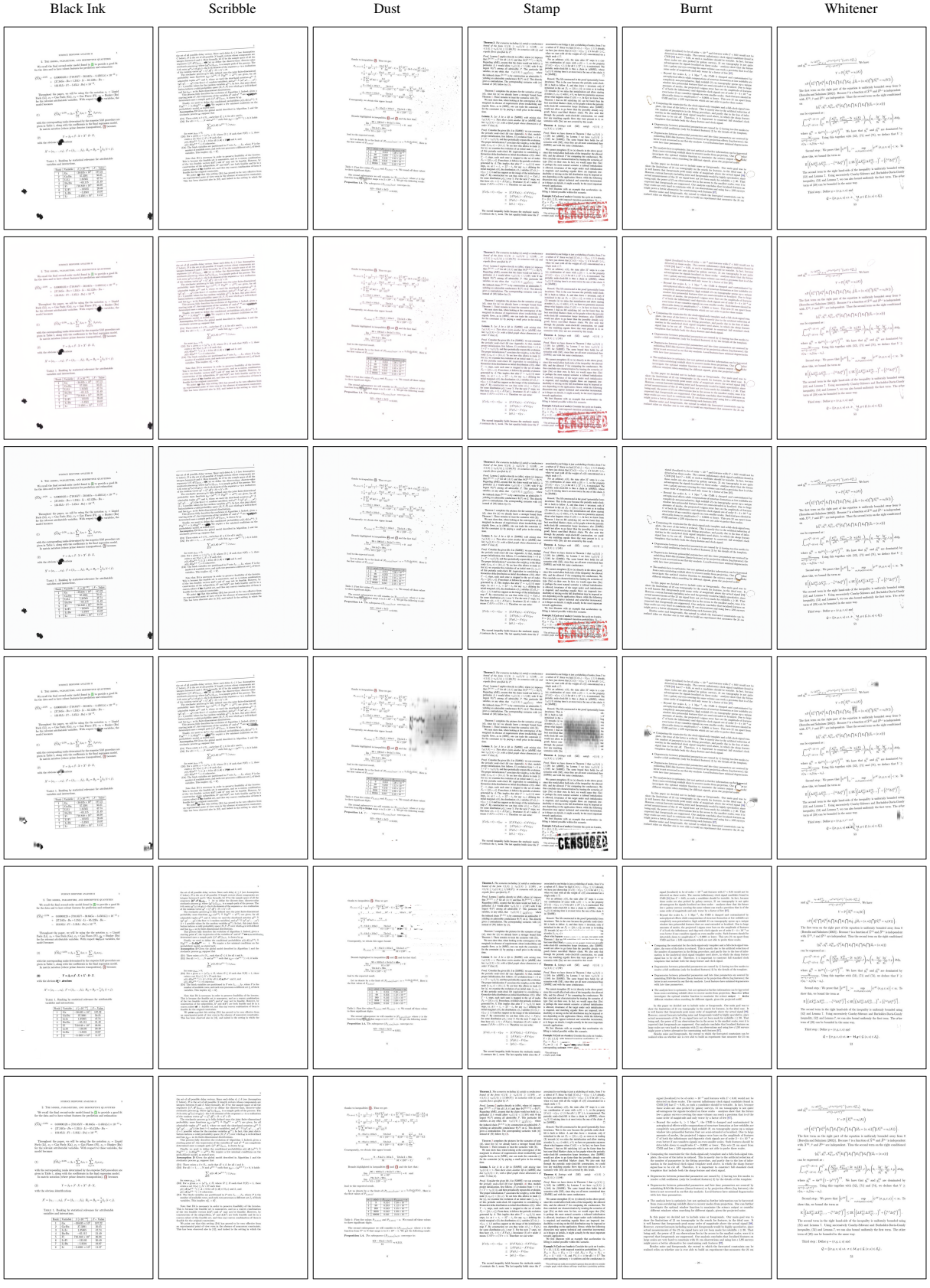

Figure 10. Qualitative comparison of document restoration methods across occlusion types.