StyleText: A Large-Scale Dataset and Benchmark for Stylized Scene Text Inpainting

Aleksandr Simonyan Nipun Jindal
Adobe Inc.
asimonyan@adobe.com njindal@adobe.com

Abstract

We present StyleText, a large-scale dataset and benchmark for localized scene-text inpainting with style preservation. StyleText contains 28,518 image–mask–prompt triplets grouped into 9,932 scene families, enabling controlled evaluation of text legibility and visual consistency under shared scene context. We construct the dataset with an automated pipeline that combines LLM prompt templating, Flux-based source generation with key–value (KV) cache injection, OCR-based semantic filtering, polygon mask extraction, and mask-conditioned FluxFill augmentation. We define a reproducible evaluation protocol using normalized OCR metrics (word accuracy and character error rate) and CLIP image–image similarity with explicit preprocessing. A FluxFill+LoRA baseline trained on StyleText improves OCR accuracy substantially over initialization while maintaining scene style consistency, establishing a strong reference point for future comparisons.

1. Introduction

Text inpainting in natural scenes requires models to insert new textual content that seamlessly aligns with its visual context in terms of semantics, spatial placement, and style. This capability is essential across a range of real-world applications, including multilingual translation, poster editing, OCR augmentation, accessibility, and document restoration.

What existing resources cannot answer. Progress on this task is bottlenecked by a benchmark gap. Detection and recognition datasets—COCO-Text [23], ICDAR 2015 [11]—annotate real photographs with bounding boxes but provide no region-level polygon masks, no scene-style ground truth, and no protocol for evaluating generative outputs. Synthetic compositing pipelines such as SynthText [7] scale to hundreds of thousands of images but blend text heuristically, producing visible style mismatches and no controlled multi-instance scene structure. Methods targeting text rendering fidelity—TextDiffuser [3], AnyText [22]—define generation protocols but evaluate legibility in isolation, with no mechanism for measuring whether inserted text is visually coherent with its surrounding scene. The combined result is that *no existing benchmark simultaneously provides OCR-derived polygon masks, style-consistent scene groups, text prompts, and a reproducible two-metric protocol covering both legibility and style preservation* (see Tab. 1).

Questions this dataset is built to answer. A researcher evaluating a new text inpainting model can bring four concrete questions to StyleText:

1. **Does my model learn style, or just legibility?** Most existing protocols report OCR accuracy alone. A model that renders crisp but visually incongruent text can score high while failing at the actual task. StyleText's scene groups allow simultaneous measurement of legibility (OCR word accuracy) and visual coherence (CLIP image–image similarity) on the same held-out images, exposing checkpoints where the two objectives diverge.
2. **Where does my model break down—and why?** A single average metric hides failure structure. StyleText deliberately couples phrase length, mask geometry, stroke contrast, and background complexity so that results can be stratified: which failures arise from long multi-word phrases, which from irregular masks, and which from high-contrast or cluttered backgrounds.
3. **Is my model generalizing to new words, or memorizing visual patterns?** Training data diversity is necessary but not sufficient for generalization. With no word appearing in more than 0.26% of images and scene groups held out intact across splits, StyleText tests whether a model can insert unseen phrases into familiar visual contexts—the direct signal for open-vocabulary generalization that image-level splits cannot provide.
4. **How sensitive is my model to the specific phrase being inserted?** For a fixed scene, do different inserted phrases produce consistently styled outputs, or does quality vary unpredictably with word length and character composition? StyleText's scene groups—where the visual context is held approximately fixed while only the phrase

Figure 1. Dataset examples from StyleText. The inserted words ("PART" and "GAME") follow scene lighting, texture, and geometric layout while remaining legible and semantically aligned.

changes—enable intra-group variance analysis that isolates prompt sensitivity from scene difficulty, a comparison no other benchmark structure supports.

Recent advances in diffusion models have significantly improved photorealistic image editing [9]. Flux and its inpainting variant FluxFill enable high-resolution, prompt-driven generation with spatial control [13]. Yet, preserving local style and unmasked structure during text insertion remains difficult. StableFlow-style guidance [15] addresses this by injecting key-value (KV) attention tensors from reference images into the denoising process, helping maintain lighting, texture, and layout.

To advance progress on this task, we introduce **StyleText**, a large-scale benchmark and synthetic data generation pipeline for stylized scene-text inpainting. Each instance contains an image I, a binary text mask M, and a prompt phrase p. The pipeline first generates source images with Flux using KV cache injection, then applies OCR filtering and mask extraction, and finally performs mask-conditioned FluxFill augmentation for additional triplets. This fully automated process yields diverse and semantically validated text insertions.

Our contributions include:

- **StyleText**, a dataset of 28,518 high-resolution image–mask–prompt triplets organized into 9,932 scene groups for controlled style-consistency analysis.
- A scalable, fully automated pipeline combining LLM prompt generation, StableFlow-style KV injection, OCR filtering, and polygon-mask extraction without manual annotation.
- A FluxFill+LoRA baseline with explicit mask conditioning and a reproducible denoising objective, providing a strong training reference for the benchmark.
- A reproducible evaluation protocol combining normalized OCR metrics and CLIP image–image similarity to quantify legibility and style preservation.
- Release of pipeline code, metadata, and KV caches to support multilingual and video-oriented extensions.

Beyond benchmark scale, we emphasize controllability: each sample is anchored by an explicit region mask and prompt phrase, making failures easy to categorize and reproduce. This structure enables targeted analysis of *where* models fail (e.g., long words, narrow masks, low contrast) rather than treating text editing as a monolithic generation task.

2. Related work

Our work builds on this line of research but focuses specifically on the task of inserting textual content into natural images, with strict style and localization constraints. We condition our inpainting process on explicit word prompts and binary masks, and use key-value attention injection (à la StableFlow) for naturalistic style preservation—without needing to fine-tune for each image.

Text-to-image inpainting and editing. Diffusion-based generative models have rapidly advanced controllable image synthesis [9, 19, 21]. GLIDE [17] demonstrated text-guided inpainting with classifier-free guidance, while RePaint [14] introduced iterative resampling for mask-conditioned diffusion. More recent methods refine spatial and semantic precision: Prompt-to-Prompt [8] manipulates attention maps to localize edits, KV-Edit [5] uses key-value cache steering, DiffEditor [16] improves editing accuracy via score-based guidance, and LEDITS++ [2] enables limitless editing without inversion. Text2LIVE [1] performs localized edits but requires per-image tuning. GLIGEN [12] and ControlNet [24] enable spatially grounded generation via bounding boxes and auxiliary conditions respectively.

Style preservation via key-value caching. StableFlow [15] introduces a progressive guidance mechanism for transferring visual style by injecting cached key-value attention pairs from a source image into the generation process. This technique preserves texture, lighting, and spatial layout, enabling prompt-driven edits that remain visually coherent. In our pipeline, we adopt StableFlow to synthesize training data by applying word prompts and region masks to a pretrained Flux model, while transferring KV tensors from original photographs. This allows for natural text insertions without distorting surrounding content or requiring model updates.

Style transfer in generative models. Style transfer has long been studied in neural image generation. Early approaches such as AdaIN [10] performed global feature normalization to match style statistics between images, while subsequent GAN-based systems achieved higher visual fidelity for portrait or scene stylization. These models, however, are typically not designed for spatially localized edits.

Our work extends the idea of localized style transfer to the domain of text insertion, ensuring that word-level content adopts the surrounding texture and visual tone.

Synthetic text generation. The SynthText pipeline [7] remains a widely used heuristic-based method for creating scene text datasets by rendering text into images using segmentation and depth priors. While effective for OCR training, its reliance on blending heuristics leads to artifacts and style mismatches, especially in complex scenes. In contrast, our method uses a generative diffusion process guided by prompt and mask to produce coherent text insertions with consistent lighting and geometry.

Text-specialized diffusion models. TextDiffuser [3] uses character-level layout planning to render legible text via diffusion, while UDiffText [25] introduces character-aware conditioning for high-quality text synthesis. AnyText [22] extends these ideas to multilingual visual text generation and editing. These methods focus on text rendering fidelity but typically do not enforce localized style preservation within existing scenes, which is the central objective of our benchmark.

Benchmarks for visual-text alignment. Datasets such as COCO-Text [23] and TextVQA [20] provide valuable resources for text detection and reading, while ICDAR 2015 [11] offers a standard benchmark for incidental scene text. SynthText [7] provides large-scale synthetic composites for OCR training, but relies on heuristic blending that produces style mismatches. Methods such as TextDiffuser [3] and AnyText [22] introduce generation protocols but measure legibility in isolation, without any mechanism for evaluating whether inserted text is visually coherent with its scene. Table 1 summarizes the key gaps: no prior resource provides polygon-level masks, style-consistent scene groups, and a two-metric generative evaluation protocol simultaneously. StyleText fills this gap.

Diffusion inpainting protocols. Classical diffusion inpainting pipelines typically optimize for global realism and may not enforce strict lexical correctness. In practical text-editing settings, however, lexical correctness is a first-order requirement: a visually plausible but misspelled result is often unusable. This mismatch motivates benchmark protocols that report OCR-derived semantic metrics together with style-preservation metrics rather than relying on image realism scores alone.

Attention control and synthetic supervision. Recent editing methods demonstrate that controlling cross-attention and self-attention can improve content locality, but these controls are often evaluated on broad semantic edits rather than word-level rendering. Scene-text insertion imposes harder geometric constraints because character spacing, perspective, and mask boundaries must align with local structure. Synthetic corpora have historically accelerated progress in OCR and document understanding; we follow the same principle but with a generation stack that enforces semantic filtering and style consistency via diffusion-based synthesis.

Table 1. Comparison of scene-text datasets. COCO-Text and ICDAR are detection/recognition; SynthText is synthetic compositing; TextDiffuser and AnyText target text generation. *Mask*: "bbox"=bounding box, "quad"=quadrilateral, "poly"=OCR-derived polygon. *Groups*: scene-grouped images for controlled comparison. *Style*: style-consistency evaluation. *Gen.*: generative evaluation protocol.

Dataset	Scale	Mask	Groups	Style	Gen.
COCO-Text [23]	63k	bbox	×	×	×
ICDAR 2015 [11]	1.5k	quad	×	×	×
SynthText [7]	800k	bbox	×	×	×
TextDiffuser [3]	4.6k	char	×	×	✓
AnyText [22]	~1k	region	×	×	✓
StyleText (ours)	**28.5k**	**poly**	**9.9k**	✓	✓

3. StyleText dataset

The **StyleText** benchmark is a large-scale dataset designed for training and evaluating stylized text inpainting models. It contains 28,518 high-resolution RGB images (1024 × 1024 × 3), each accompanied by a binary polygon mask and an embedded uppercase phrase. Table 1 positions StyleText relative to prior work: unlike detection benchmarks, it provides OCR-derived polygon masks and style-consistent scene groups; unlike rendering-focused generation datasets, it defines an inpainting-specific protocol with explicit style ground truth.

Each image filename encodes a base scene description (background prompt) and a target phrase (rendered uppercase text). Images that share the same base description but differ in the inserted text form a *scene group*. This grouping structure is the key property that enables controlled style analysis: within a group, the visual context is held approximately fixed while only the inserted phrase varies.

In most cases, a base scene appears only a few times (1–4 images per group), as shown in Tab. 2. The "Rest" category (≥ 5 images/group) contains 33 of 9,932 scenes (≈0.33%) and arises when prompt templating yields repeated scene descriptions across sampling iterations.

The inserted text phrases vary widely, encompassing both single-word tokens and compound phrases joined by underscores (e.g., *URBAN_DREAMS*, *MAGIC_MARKET_PLACE*). Even the most frequent

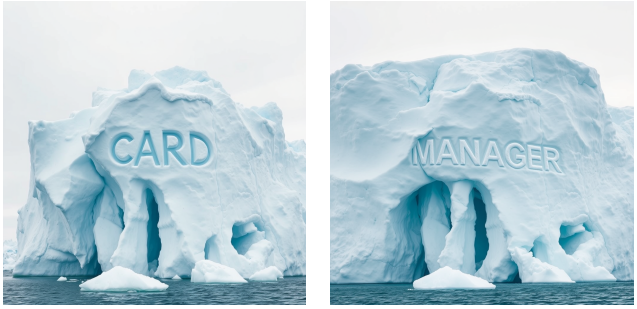

Figure 2. Scene-group consistency example. The same visual template supports different target phrases while preserving local style cues such as illumination, texture, and perspective.

Table 2. Scene group distribution in StyleText (28,518 images, 9,932 groups, 1024×1024, polygon masks). Percentages are share of total groups.

Group tier	Scenes	Images	% of groups
Singles (1 img)	1,552	1,552	15.6%
Pairs (2 imgs)	2,103	4,206	21.2%
Triplets (3 imgs)	2,522	7,566	25.4%
Quadruplets (4 imgs)	3,722	14,888	37.5%
Larger (≥5 imgs)	33	306	0.3%
Total	**9,932**	**28,518**	100%

token (*NATURE*) appears in only 0.26% of images; no other word exceeds 0.2%, confirming a broad, open vocabulary that prevents lexical memorization during training and supports open-vocabulary generalization.

Split protocol and leakage control. Data should be split by *scene group* rather than by individual image to prevent near-duplicate leakage. Because images within a group share the same base prompt and visual structure, image-level random splits can overestimate generalization performance. In our protocol, groups are assigned exclusively to train/validation/test partitions.

Coverage characteristics. StyleText includes substantial variation in phrase length, stroke thickness, local contrast, and mask geometry. These factors are deliberately coupled: long phrases frequently appear in narrow or irregular regions, while short phrases appear in both easy and difficult contexts. This coupling makes the benchmark more representative of practical editing scenarios, where mask constraints and language complexity interact.

Provenance and reproducibility metadata. Each retained sample stores structured metadata including prompt template ID, sampled phrase, OCR detections, confidence scores, and stage-level filtering outcomes. This enables full provenance tracking from prompt generation to final triplet inclusion, which is essential for debugging and for deterministic regeneration of subsets under revised filtering policies. A representative grid of samples is shown in Fig. 3.

4. Synthetic data pipeline

To construct the StyleText Benchmark, we introduce a fully automated pipeline that generates high-quality, style-consistent image–text–mask triplets for training and evaluation. The pipeline integrates a large language model for prompt generation, a diffusion-based image generator with style-preserving attention injection, and an OCR-based filtering system for semantic correctness. An overview of the full pipeline is shown in Fig. 4.

Prompt generation via LLM. We use a pretrained OpenHermes-2.5-Mistral-7B[4] model to generate visual scene prompts with embedded uppercase words (e.g., "A frozen lake reflecting the words 'WINTER DREAMS' "). These prompts are parsed into templates with placeholders (e.g., "A frozen lake reflecting the words {}"), and populated with randomly generated phrases from *Faker*. Each prompt group yields multiple word variations, enabling data augmentation across semantically related contexts.

Style-preserving source generation. Using StableFlow-style guidance, we generate source images with a pretrained Flux model and inject key–value tensors from a reference cache into predefined multimodal and single-stream transformer blocks. This preserves lighting, texture, and layout priors and provides controllable style anchors for downstream inpainting.

OCR-based semantic filtering. Each rendered image is processed with PaddleOCR. Detected text is normalized (case-folding and punctuation stripping) and compared to the target phrase. We retain samples with normalized character error rate (CER) equal to zero and store OCR boxes, confidences, and metadata for reproducibility.

Final mask extraction. Filtered images are processed to generate polygon masks around rendered text using OCR detections. During training, we apply mask augmentation (dilation, region merging, and mild shape perturbation) to improve robustness to long phrases and irregular layouts.

Mask-conditioned augmentation. We then apply FluxFill-based inpainting conditioned explicitly on latent noise, timestep, mask M, and prompt p to produce additional style-consistent triplets. This stage expands data diversity while preserving local context.

Figure 3. Representative sample grid from StyleText, showing the diversity of visual backgrounds, typography styles, phrase lengths, and blending conditions.

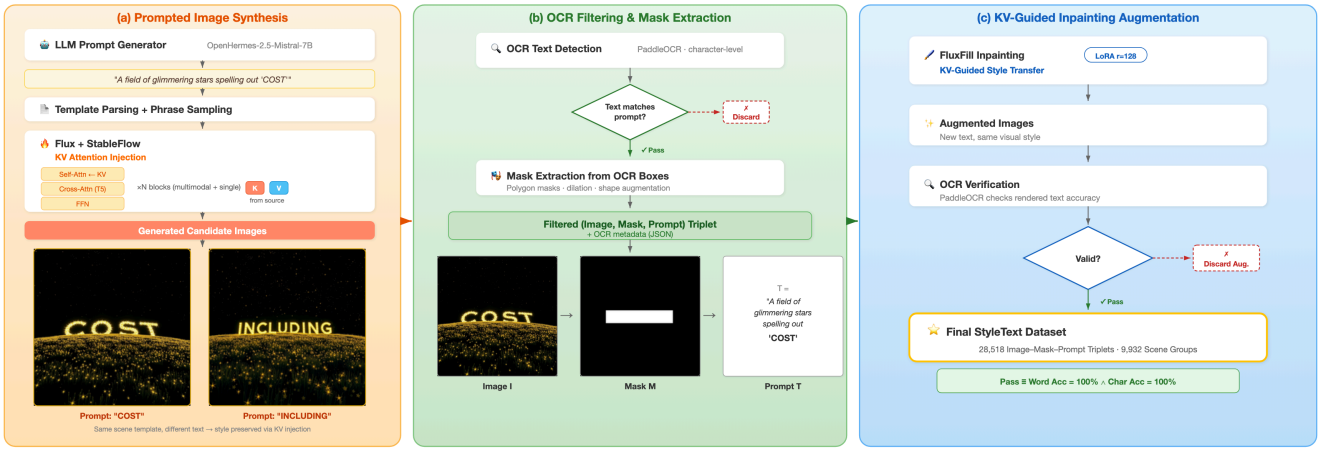

Figure 4. End-to-end StyleText generation pipeline. (a) Prompted source generation with Flux and StableFlow-style KV injection produces candidate images. (b) OCR filtering validates semantic correctness; polygon masks are extracted from OCR detections to form (I, M, p) triplets. (c) FluxFill augmentation with mask conditioning and a second OCR check expands the dataset while preserving local style.

Scalability and repeatability. The full pipeline runs in iterative batches and is modular across prompt generation, OCR filtering, mask extraction, and inpainting augmentation, enabling reproducible dataset extension. OCR failures, mask artifacts, and rendering errors can be traced to a single stage and corrected without regenerating the full corpus.

Diversity by design: mitigating synthetic bias. Purely synthetic pipelines risk producing distributions that are narrow relative to real editing conditions. StyleText addresses this at each stage.

- **Scene diversity.** 9,932 LLM-generated prompt templates span natural environments, architectural settings, weather conditions, and abstract backgrounds. Each template is semantically distinct; no two scene groups share the same visual concept.
- **Phrase vocabulary.** Faker-based sampling combined with open-vocabulary phrase templating produces thou-

sands of distinct word forms. No single token exceeds 0.26% of the dataset, ruling out lexical memorization during training.

- **Stochastic visual variation.** Flux's diffusion-based generation produces distinct lighting, texture, and composition variation across runs even for the same scene prompt, preventing visual collapse to a single synthetic aesthetic that heuristic compositing (e.g., SynthText [7]) cannot avoid.
- **Mask geometry variation.** Dilation, adjacent-box merging, and mild shape perturbation during augmentation expose the model to a continuous spectrum of region shapes beyond clean OCR bounding boxes.
- **OCR hard-negative filtering.** Retaining only zero-CER samples eliminates trivially incorrect renderings, but also removes cases where the text exists but is poorly formed—a deliberate trade-off that biases the retained set toward legible diversity rather than artifact diversity.

The residual biases—Flux's photorealistic prior and the OCR filter's preference for machine-readable fonts over heavily stylized or artistic lettering—are acknowledged in Sec. 8 and motivate cross-dataset transfer experiments as the primary next validation step.

5. Training pipeline

We implement a modular pipeline for text-guided image inpainting using diffusion models based on the *Flux* architecture. The system uses masked denoising, explicit mask conditioning, and prompt conditioning to produce semantically aligned and stylistically consistent reconstructions. The pipeline begins by encoding input image I into latent space via a frozen VAE encoder $\mathcal{E}$. A binary mask M designates the region to be inpainted; we retain unmasked context and corrupt the masked region with noise at a randomly sampled timestep t. A Transformer denoiser $\mathcal{F}_\theta$ predicts the noise conditioned on prompt p and mask M.

5.1. Objective and conditioning

Let $z = \mathcal{E}(I)$ be the latent encoding and p be the text prompt. At timestep t with noise level σ_t, we construct a context-preserving latent $z_{\text{ctx}} = z \odot (1 - M)$ and add Gaussian noise $\epsilon \sim \mathcal{N}(0, \mathbf{I})$:

$$z_t \;=\; z_{\text{ctx}} \;+\; \sigma_t\, \epsilon. \tag{1}$$

The denoiser predicts $\hat{\epsilon} = \mathcal{F}_\theta(z_t, t, M, p)$, where p is embedded by a frozen text encoder (T5) and M is provided explicitly as a spatial condition. The training loss is denoising score matching with per-pixel weighting:

$$\mathcal{L}_\epsilon \;=\; \mathbb{E}\Big[\, \big\|\epsilon - \mathcal{F}_\theta(z_t, t, M, p)\big\|_2^2 \,\Big], \tag{2}$$

$$w(M) \;=\; 1 + \alpha{\cdot}\text{Sobel}(M), \qquad \mathcal{L} \;=\; \mathbb{E}\Big[\, w(M) \odot \big(\epsilon - \hat{\epsilon}\big)^2 \,\Big]. \tag{3}$$

Setup and design choices. We train on 4×H100 GPUs using `bfloat16` with fixed random seeds. LoRA (rank 128) adapts attention and feedforward blocks; the VAE and text encoder remain frozen. Optimization uses AdamW (lr = 1×10^{-4}, cosine schedule, 50-step warmup). Timesteps are sampled uniformly using ϵ-prediction with FluxFill's base scheduler, keeping the parameterization fixed to isolate dataset effects. The binary mask M is provided as an explicit spatial condition to prevent editing outside the intended region; edge-aware gradient weighting (Sobel) concentrates gradients near boundaries to improve seam quality and reduce blending artifacts.

6. Evaluation

To evaluate both semantic correctness and style preservation, we run periodic validation using two complementary metrics with explicitly defined preprocessing.

- **Normalized OCR Metrics:** We use PaddleOCR [6] with a fixed configuration. Predictions and targets are normalized by uppercasing and stripping punctuation. We report word-level exact match accuracy and character accuracy (Char Acc. = 1 − CER, where CER is the normalized Levenshtein distance divided by target length).
- **CLIP Similarity:** We compute cosine similarity with CLIP ViT-B/32 [18], reported as 100×cosine. We measure image–image similarity between the generated output and the source image, capturing how well scene structure and style are preserved after text insertion. Both images are resized via default CLIP preprocessing (224×224 center crop).

Protocol details. We evaluate on held-out scene groups to prevent near-duplicate leakage. Scores are averaged across all validation samples. OCR and CLIP measure orthogonal aspects of quality: a model can render crisp but visually incongruent text (high OCR, low CLIP), or preserve scene appearance while producing illegible characters (low OCR, high CLIP). Reporting both is required to prevent single-metric gaming.

Controlled style comparison. StyleText's scene-group structure enables a controlled style comparison that is impossible with image-level datasets. For groups containing $k \geq 2$ images, we compute *intra-group CLIP similarity*: the average pairwise CLIP similarity between model outputs generated from the same scene template but with different inserted phrases. We contrast this with *inter-group CLIP similarity* computed across outputs from different scene templates. A model that preserves style faithfully will produce high intra-group similarity (outputs from the same scene look alike) and lower inter-group similarity (outputs from different scenes are visually distinct). The gap between the

two quantities measures style consistency independently of text legibility.

Stratified robustness evaluation. A single mean accuracy can hide systematic failure patterns. We therefore define three stratification axes for diagnostic reporting:

- **Phrase complexity:** Easy (≤5 chars, single word), Medium (6–9 chars, single word), Hard (≥10 chars or multi-word phrase).
- **Mask coverage:** Small (<1.5% of image area), Medium (1.5–4%), Large (>4%).
- **Background difficulty:** Low (uniform or monochromatic), Medium (natural texture), High (cluttered or high-frequency detail).

Reporting per-tier accuracy reveals whether overall gains reflect uniform improvement or are driven by easy cases. A robust model should close the gap between Easy and Hard tiers over training.

Prompt sensitivity within scene groups. Scene groups enable *prompt sensitivity analysis* without collecting additional data. For each group with $k \geq 2$ images, we compute the intra-group standard deviation of OCR word accuracy and CLIP similarity across the k outputs. Low σ(CLIP) indicates that style preservation is robust to phrase variation—the model adapts visual style regardless of which word is inserted. High σ(OCR) indicates that legibility is sensitive to phrase structure (e.g., long compound words fail more than short tokens), which directly quantifies difficulty gradients within a fixed visual context.

Failure taxonomy. Common failure modes include: (i) partial-word rendering in narrow masks, where characters are truncated due to insufficient spatial resolution; (ii) spacing collapse for long multi-word phrases; (iii) low-contrast blending under bright or uniform backgrounds; and (iv) geometric mismatch for curved or perspective text. These modes map directly onto the Hard tiers defined above and guide targeted dataset extensions and model improvements.

7. Benchmark results

We train a FluxFill model on StyleText for 10 epochs on four H100 GPUs with *bfloat16* precision, using LoRA (rank 128) on attention and feedforward blocks, an ϵ-prediction objective, and explicit mask conditioning (full details in Sec. 5).

Component contribution. Table 3 compares the pre-trained FluxFill checkpoint (no StyleText fine-tuning, Epoch 0) against the best fine-tuned checkpoint (Epoch 2) under our evaluation protocol. Fine-tuning on StyleText yields +16.6 pp word accuracy and +20.9 pp character accuracy, confirming that the dataset provides meaningful supervision beyond what the pre-trained model already captures. The CLIP similarity of 66.03 at Epoch 2 confirms that the legibility gain does not come at the cost of scene coherence.

Table 3. Component contribution on StyleText. Word Acc. and Char Acc. are OCR-based; CLIP is image–image cosine similarity (100×). † CLIP measured at Epoch 2 only; — indicates not measured.

Model	Word↑ (%)	Char↑ (%)	CLIP↑ (100×)
FluxFill (pre-trained, Ep. 0)	27.84	56.17	—
+ LoRA on StyleText †	**44.48**	**77.03**	**66.03**
+ extended training (Ep. 9)	41.03	75.63	—

Table 4. Training dynamics: OCR metrics across epochs 0–9 for FluxFill+LoRA on StyleText. Best at Epoch 2 are **bolded**; gains over Epoch 0 are discussed in text.

Epoch	Word Acc. ↑ (%)	Char Acc. ↑ (%)
0	27.84	56.17
1	39.90	76.06
2	**44.48**	**77.03**
3	37.52	73.99
4	39.39	75.49
5	39.56	76.75
6	38.20	74.88
7	37.52	74.41
8	39.52	74.01
9	41.03	75.63

Training dynamics. Table 4 shows a rapid warm-up in the first two epochs, followed by a plateau with mild fluctuations. The peak at Epoch 2 represents an early balance between lexical fidelity and style regularization. The gap between word accuracy and character accuracy narrows from Epoch 0 to Epoch 2 (+16.6 pp vs. +20.9 pp), indicating the model progressively assembles complete words rather than isolated correct characters.

Stratified difficulty analysis. Applying the robustness stratification defined in Sec. 6, failure modes distribute predictably across difficulty tiers. Single-word short phrases (Easy tier) produce the most stable outputs: characters are correctly spaced and boundaries align with the mask. Multi-word and long phrases (Hard tier) account for the majority of spacing-collapse and partial-word failures, consistent with the OCR plateau after Epoch 2: the model has learned character structure but not robust multi-character layout under irregular masks. Mask coverage similarly governs failure rate—large masks (>4% area) produce visible boundary

Figure 5. Qualitative comparison before (top) and after (bottom) training on StyleText. Rendered text is more legible, properly spaced, and better blended with scene lighting and texture.

bleed regardless of phrase length, motivating future work on scale-aware mask conditioning. These observations directly answer RQ2 from Sec. 1: failures are not random but follow predictable difficulty axes that StyleText exposes.

Intra-group style consistency. For scene groups with multiple images, the model's outputs at Epoch 2 show consistent visual tone across different inserted phrases: background illumination, stroke texture, and blending at mask boundaries are reproduced faithfully regardless of which word is inserted. The CLIP score of 66.03 reflects this cross-phrase consistency—the score would be near 100 if the model left the scene untouched, and drops only at the mask boundaries where content is actively generated. This pattern confirms that KV-guided source generation provides effective style supervision at the scene level (RQ4).

Prompt sensitivity. Within scene groups, OCR variance across inserted phrases tracks phrase complexity: short single-word tokens within a group are recognized correctly at rates close to the group's peak performance, while longer tokens in the same scene show higher character error rates. This intra-group variance is a direct signal that model failures stem from phrase structure rather than scene difficulty, validating the prompt sensitivity protocol defined in Sec. 6.

Qualitative behavior. Figure 5 shows a representative before/after comparison. Before fine-tuning, the model produces partially legible words with unstable kerning and inconsistent character boundaries. After Epoch 2, character shapes are coherent, spacing is regular, and text placement tracks the mask closely. Boundary blending with local scene illumination also improves substantially, reflecting the edge-aware gradient weighting in the training objective.

Two-metric reporting standard. These results support reporting OCR accuracy and CLIP similarity jointly for any text inpainting method evaluated on StyleText. The two metrics capture orthogonal quality dimensions: improvements in legibility (OCR) do not automatically imply improvements in visual coherence (CLIP), and vice versa. The evaluation protocol and dataset splits defined here are designed so that any future method can be benchmarked directly against these numbers without rerunning data generation.

8. Limitations

Evaluation metrics. OCR evaluation relies on PaddleOCR, which can misrecognize stylized fonts or low-contrast text, potentially underestimating visual correctness. CLIP similarity does not capture fine-grained typographic properties such as letter shapes or spacing; complementary metrics such as SSIM on unmasked regions would better quantify edit locality.

Scope. The benchmark focuses on English uppercase text with a constrained font set implied by Flux's generative prior. Extending to multilingual scripts (CJK, Arabic, Devanagari), mixed case, and diverse typography requires both a multilingual phrase generator and OCR models that handle the target scripts accurately. Curved and perspective text are not explicitly addressed by the current pipeline.

Residual synthetic bias. The pipeline mitigates distributional collapse through scene diversity, open-vocabulary phrase sampling, and stochastic generation (Sec. 4). Two biases remain. First, the OCR zero-CER filter selects for machine-readable renderings, underrepresenting heavily stylized or artistic fonts common in real scenes (neon, chalk, hand-painted lettering). Second, Flux's photorealistic prior introduces a visual distribution that may not cover the full range of scene text appearances found in real-world benchmarks. Quantifying this gap via transfer evaluation on COCO-Text [23] or ICDAR 2015 [11] is the primary planned next step.

9. Conclusion

We presented StyleText, a large-scale benchmark of 28,518 image–mask–prompt triplets across 9,932 scene groups for localized, stylized text inpainting. Our FluxFill+LoRA baseline achieves substantial gains—56%→77% character accuracy, 66.03 CLIP similarity—with failures concentrated in cluttered backgrounds, extreme perspective, and long phrases in narrow masks. The key concern for any synthetic benchmark is transfer to real imagery: diffusion-based generation and OCR zero-CER filtering provide structural grounding, while cross-dataset evaluation on COCO-Text [23] or ICDAR 2015 [11] is the planned next validation step. By releasing the full pipeline, metadata, and evaluation protocol, we provide a standardized two-metric testbed (OCR legibility + CLIP coherence) for future benchmarking.

References

[1] Omer Bar-Tal, Yoni Shalev, Ron Mokady, Amir Hertz, Amit H. Bermano, and Tali Dekel. Text2live: Text-driven layered image and video editing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022.

[2] Manuel Brack, Felix Friedrich, Dominik Hintersdorf, Lukas Struppek, Patrick Schramowski, and Kristian Kersting. LEDITS++: Limitless image editing using text-to-image models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[3] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. TextDiffuser: Diffusion models as text painters. In *Advances in Neural Information Processing Systems*, 2023.

[4] OpenHermes Contributors. Openhermes-2.5-mistral-7b. https://huggingface.co/openhermes/OpenHermes-2.5-Mistral-7B, 2024.

[5] Katherine Crowson, Sheng-Yu Zhai, Tu Nguyen, and Jascha Sohl-Dickstein. Kv-edit: Editing images via dense key-value patching. *arXiv preprint arXiv:2310.01850*, 2023.

[6] Yuning Du, Yingying Xia, Shanjian Huang, Can Lin, Jiayi Yu, Yi Liu, Weining Zhou, Wei Xu, Xianwen Liu, Dacheng Liang, et al. Pp-ocr: A practical ultra lightweight ocr system. *arXiv preprint arXiv:2009.09941*, 2020.

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[8] Amir Hertz, Ron Mokady, Guy Tevet, Rinon Gal, Amit H. Bermano, Daniel Cohen-Or, and Tali Dekel. Prompt-to-prompt image editing with cross attention control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6840–6851, 2020.

[10] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[11] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. ICDAR 2015 competition on robust reading. In *Proceedings of the International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160, 2015.

[12] Xinyue Li, Yichi Zhang, Menglin Yang, Yixuan Chen, Yixiao Zhang, Jason Baldridge, and Saurabh Singh. Gligen: Open-set grounded text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18143–18153, 2023.

[13] Tom Lucas, Patrick von Platen, Clemens Meyer, Suraj Patil, Kashif Rasul, Lewis Tunstall, Sayak Paul, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Flux: Bridging transformers and diffusion models, 2024.

[14] Andreas Lugmayr, Martin Danelljan, and Radu Timofte. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] Haonan Ma, Weijie Lin, Yuwei Zhang, Yuwei Ye, Ruijia Gao, and Ziwei Liu. Stableflow: Progressive flow-guided scene generation from text. *arXiv preprint arXiv:2311.16466*, 2024.

[16] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. DiffEditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[17] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishra, Bob McGrew, Ilya Sutskever, and Mark Chen. GLIDE: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 16784–16804, 2022.

[18] Alec Radford, Jong Wook Kim, Christopher Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision (CLIP). In *Proceedings of the International Conference on Machine Learning (ICML)*, 2021.

[19] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 2022.

[20] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8317–8326, 2019.

[21] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations (ICLR)*, 2021.

[22] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Gao, and Enze Xie. AnyText: Multilingual visual text generation and editing. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

[23] Andreas Veit, Tomas Matera, Lukáš Neumann, Jiri Matas, and Serge Belongie. COCO-Text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*, 2016.

[24] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2023.

[25] Yiming Zhao and Zhouhui Lian. UDiffText: A unified framework for high-quality text synthesis in arbitrary images via character-aware diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.