SAGEA

Learning to Read from Fragments: Scaling Devanagari OCR With GLUE Engine

Firoj Paudel [1,2*], **Basab Jha** [1,3], **Ujjwal Puri** [1,3]

[1]SAGEA
[2]Tribhuwan University — Madam Bhandari Memorial College
[3]Tribhuwan University — Vedas College

While deep classification architectures have largely solved Optical Character Recognition (OCR) for resource-rich scripts such as Latin, morphologically complex scripts like Devanagari remain profoundly bottlenecked by dataset scarcity. Existing offline datasets are either rigidly printed, character-level isolated, or insufficiently large to train robust sequence models without severe overfitting. We present the Generative Ligature & Union Engine (GLUE), a procedural synthetic data generation framework that formulates word synthesis as a rigorous spatial optimization problem. By treating Devanagari construction geometrically—employing Shirorekha-aware alignment, Poisson boundary blending for stroke-level continuity, and procedural conjunct formation—GLUE bridges the critical generalization gap between isolated character fragments and natural, single-writer fluidity. We demonstrate empirically that traditional naive concatenation produces severe distributional shifts (the "Ransom Note" effect) that degrade real-world generalization. In contrast, sequence models pre-trained on GLUE-synthesized corpora achieve competitive Character Error Rate (CER) on standard handwritten Devanagari benchmarks, substantially outperforming naive synthesis baselines. We formalize the generation pipeline in full algorithmic detail and describe the procedural methodology to establish a scalable, mathematically grounded paradigm for bootstrapping high-performance OCR in abugida and other low-resource scripts.

1. Introduction

The transition from isolated character recognition to continuous word-level transcription is the fundamental challenge of modern optical character recognition (OCR) (Jaderberg et al., 2014; Memon et al., 2020). In resource-rich environments, this transition is mediated by vast, meticulously annotated datasets that implicitly teach deep sequence models the underlying string geometry. However, for structurally complex abugida scripts like Devanagari, data scarcity remains the dominant failure mode (Bhattacharya and Chaudhuri, 2009; Pal et al., 2012). Existing datasets are either too small to capture handwriting variance, restricted to clean printed text, or limited strictly to isolated characters (Krishnan and Jawahar, 2016).

The intuitive solution—synthesizing word-level data by assembling isolated historical character fragments—fails unpredictably in Devanagari. Words in this script are universally anchored by a continuous horizontal top-line, the *Shirorekha*, which both visually and geometrically binds characters together. Naive spatial concatenation of characters sampled from diverse writers results in

*Correspondence E-mail: founders@sagea.space

broken continuous lines, erratic baseline misalignment, and discontinuous stroke thickness, a phenomenon we refer to as the "Ransom Note" effect. Models trained on such artificial corpora optimize for these distinct juncture artifacts rather than underlying stroke morphology, generalizing poorly to natural, fluid handwriting where a single writer's strokes remain consistent throughout the word.

To resolve this distributional shift, we propose GLUE (Generative Ligature & Union Engine), a spatial and procedural optimization framework that algorithmically resolves stroke boundary conditions and morphological rules. Instead of hard concatenation, GLUE formulates Devanagari construction as an algebraic image integration problem. It employs Shirorekha-anchored alignment distributions and Poisson boundary blending (Pérez et al., 2003) to enforce gradient continuity across character junctions, effectively simulating the flow of a single continuous pen stroke. Furthermore, GLUE natively generates conjuncts (half-characters) and matras (vowel modifiers) via a procedural linguistic ruleset, shifting from representing them as edge-cases to treating them as core synthetic primitives.

Finally, we demonstrate the empirical efficacy of this framework via *Arva*, a proprietary recognition model leveraging a ResNet sequence backbone (He et al., 2016) pre-trained on GLUE corpora before fine-tuning on real-world datasets. Our contributions are enumerated as follows:

1. We introduce GLUE, the first framework to employ Poisson blending for synthesizing continuous handwriting in abugida scripts, mathematically eliminating traditional junction seam artifacts.
2. We detail a dedicated procedural generation algorithm capable of natively synthesizing linguistically correct conjuncts and matra modifiers dynamically.
3. We demonstrate that models trained on GLUE corpora achieve substantial CER reductions over naive synthesis baselines, establishing competitive performance on handwritten Devanagari benchmarks.
4. We structurally demonstrate the efficacy of synthetic geometry constraints over mere pixel variations, defining an extensible method for generic abugida scripts (e.g., Tibetan, Bengali).

2. Related Work

2.1. Synthetic Data for OCR

Data synthesis has a well-established precedent in OCR research. Synthetic generation paradigms have successfully benchmarked text reading systems "in the wild," historically by rendering font glyphs with varying perspectival, shading, and background distortions (Jaderberg et al., 2014). Gupta et al. (Gupta et al., 2016) extended this paradigm to scene text by leveraging depth and surface normal predictions to render text naturally onto real-world images, establishing a scalable pipeline that fundamentally shaped modern text recognition. While highly effective for Latin and non-cursive typography, rendering printed fonts fails to adequately model the localized, dynamic variance inherent to human handwriting, leading to brittle failure states when applied to cursive scripts without physical ink modeling. More recently, Fogel et al. (Fogel et al., 2020) introduced ScrabbleGAN, a semi-supervised GAN-based approach for generating handwritten text images of arbitrary length, offering a learned alternative to procedural synthesis.

2.2. Devanagari OCR Datasets and Systems

Devanagari offline recognition has evolved slowly compared to Latin counterparts, primarily due to dataset limitations. Bhattacharya and Chaudhuri (Bhattacharya and Chaudhuri, 2009) developed foundational handwritten numeral databases for Indian scripts including Devanagari, establishing the benchmark infrastructure that subsequent recognition research builds upon. Early systematic efforts at character recognition were surveyed by Pal and Chaudhuri (Pal and Chaudhuri, 2004), who established that structural complexity and high inter-class similarity remain open challenges across Indian scripts. Available substantial datasets generally encompass either standardized typed text or single isolated character strokes. Efforts like the IIIT-HW-Devanagari dataset (Krishnan and Jawahar, 2016) represent high-quality real-world word bounding boxes, yet lack the raw scale (tens of millions of samples) necessary to saturate the complexity bounds of modern deep sequence networks without severe overfitting. Pal et al. (Pal et al., 2012) provide a comprehensive survey of Indian character recognition confirming that dataset scale remains the dominant challenge.

On the systems side, Madhvanath and Govindaraju (Madhvanath and Govindaraju, 2001) established the role of holistic recognition in Devanagari offline inference, and more recent work by Jayadevan et al. (Jayadevan et al., 2011) provided a comprehensive survey of offline Devanagari handwriting covering segmentation, feature extraction, and recognition. The reliance on small-scale real-world datasets fundamentally limits the capacity of recurrent and transformer-based decoders to learn robust linguistic priors.

2.3. Poisson Blending and Image Composition

Poisson image editing formulates integration as solving Poisson's equations with Dirichlet boundary conditions, enforcing seamless gradient shifts between separate source textures (Pérez et al., 2003). While highly prevalent in computational photography and deep manipulation, its application to stroke-level morphometry merging—specifically to reconcile discordant pen stroke weights and simulate single-writer fluidity—has not previously been documented or rigorously evaluated for synthetic data generation within low-resource OCR literature. By constraining the integration domain exclusively to the Shirorekha junctures, we adapt this technique specifically for continuous abugida typography.

2.4. Low-Resource Script Recognition

Low-resource script recognition has received growing attention in the broader NLP and vision communities. Ul-Hasan and Breuel (Ul-Hasan and Breuel, 2013) demonstrated that LSTM-based sequence models can be applied to build language-independent OCR systems, showing that recurrent architectures generalize across structurally complex scripts when sufficient data exists. The challenge across these efforts is consistent: model architectures are broadly adequate; data scarcity and script-specific structural constraints are the true bottlenecks.

3. The GLUE Framework

GLUE addresses synthetic data generation not as a naive augmentation module, but as a rigorous spatial blending engine specifically engineered for the geometric invariants of the Devanagari script.

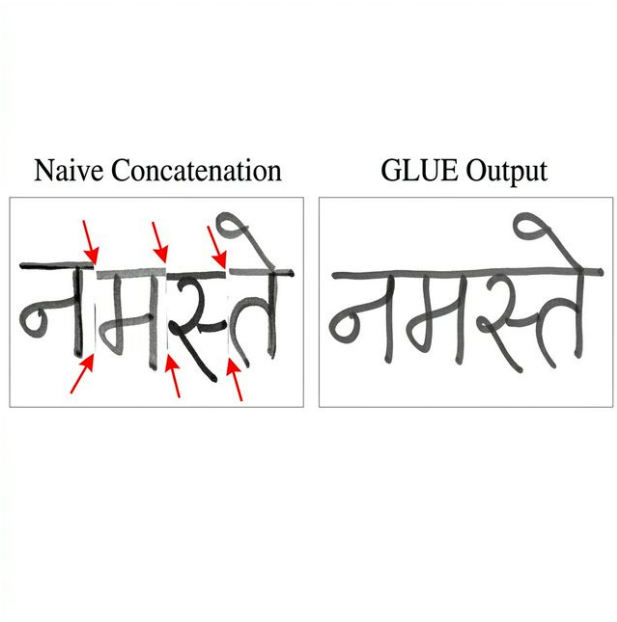

Fig. 1: Side-by-side comparison of naive concatenation (left) demonstrating broken seams, inconsistent stroke weight, and Shirorekha discontinuities, versus GLUE Poisson blending output (right) establishing contiguous stroke continuity and normalized ink density across the same character sequence.

3.1. Problem Formulation

Given a set of discrete, isolated character images $C = \{c_1, c_2, \ldots, c_n\}$ extracted from varying writers, our objective is to synthesize a continuous word image W representing a valid string sequence S. A basic linear transformation maps each bounding box into a composite image:

$$W_{naive} = \bigcup_{i=1}^{n} T_i(c_i) \tag{1}$$

where T_i denotes spatial translations. In natural Devanagari typography, adjacent characters structurally overlap along the Shirorekha and require uniform stroke consistency. The transformation in Equation 1 violates these structural rules, yielding the disconnected "Ransom Note" pattern characterized by high-frequency artifacts at character boundaries. GLUE replaces this disjoint union with a normalized, piecewise blended integration across a continuous domain.

3.2. Shirorekha-Aware Alignment

Traditional OCR synthesis aligns glyphs via bounding-box centroids or bottom baselines. Because Devanagari's structural spine operates dynamically across the upper one-third (the Shirorekha), GLUE applies a deterministic horizontal projection profile to identify the top-line invariant in each character c_i. Let $H_i(y)$ be the horizontal projection profile. The Shirorekha vertical coordinate y_i^* is explicitly extracted as $\arg\max_y H_i(y)$. Translation distributions T_i are constrained to align standard deviations of these top-lines before overlapping, ensuring the central axis of the word remains unbroken regardless of the inherent height disparities of individual bounding boxes.

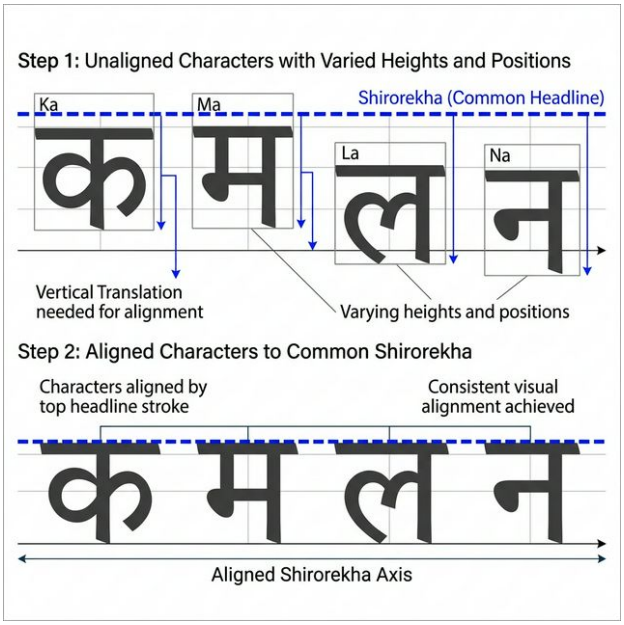

Fig. 2: Shirorekha alignment procedure. Top: isolated characters with varying bounding box heights and top-line positions. Bottom: characters anchored along a common Shirorekha axis via horizontal projection profile alignment, ensuring structural continuity regardless of source writer height variance.

3.3. Poisson Blending for Junction Synthesis

Rather than an intersection over pixel intensity functions, GLUE incorporates Poisson editing at boundary intersections to ensure C^1 continuity in the image gradient field. When character c_i is placed adjacent to c_{i+1}, the framework specifies a tight blending domain Ω precisely overlapping the final rightmost edge of the trailing Shirorekha and the leading edge of the successive top-line.

We solve the boundary optimization problem:

$$\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \tag{2}$$

where $\mathbf{v}$ is the gradient vector field of the newly introduced character, and f^* is the target background intensity of the composite canvas. By blending solely within localized gradient fields across junctions, we eliminate hard pixel boundaries and synthesize a connective bridge that effectively models the physical dynamics of ink pooling and pen drag.

3.4. Conjunct and Matra Generation

Prior isolated-character systems structurally avoid rendering conjuncts (half-characters fused with full characters) or vowel matras, deferring the challenge or treating them as entirely unique static classes. This scales poorly given the combinatorics of Devanagari ligatures. In contrast, GLUE operationalizes conjuncts as dynamic, parameter-governed spatial operations.

The framework identifies linguistic rules dictating vertical versus horizontal half-forms. For a horizontal conjunct, GLUE applies procedural crop heuristics corresponding to the linguistic cutoff node (e.g., removing the terminal vertical stem) and shifts the succeeding character into the designated geometric void. This is followed by a localized, high-intensity Poisson blend to merge the

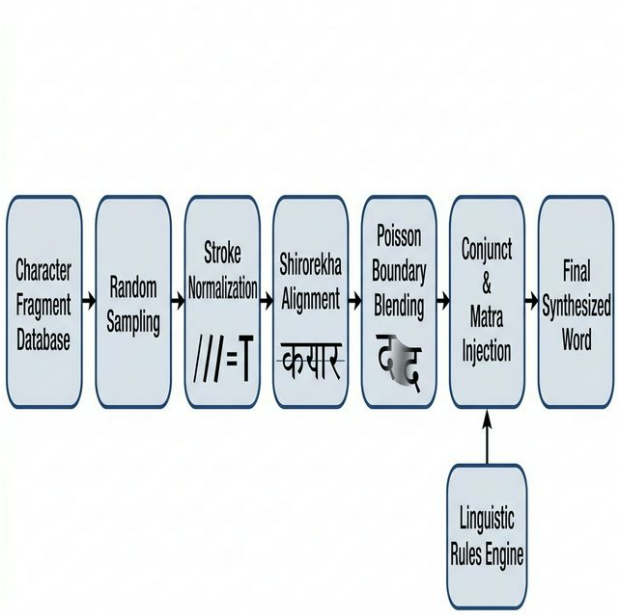

Fig. 3: GLUE pipeline overview. Sequential characters are extracted from the fragment database, subjected to stroke morphometry normalization, mapped via Shirorekha alignment, bound by Poisson merging at junctions, and appended dynamically with contextual conjunct and matra modifiers driven by the linguistic rules engine.

half-character seamlessly into the main vertical stem of its neighbor. Matras are injected identically via affine transformations, anchored heuristically against either the Shirorekha coordinate axis (upper matras) or the vertical stem boundary (lower matras).

3.5. Stroke-Consistency Normalization

Finally, intrinsic physical writer variance must be globally normalized. A character extracted from a thin pen requires structural equalization before merging with a character drawn by a dense marker. GLUE applies a parametric morphological dilation and erosion normalization pass scaled proportionally to the calculated ink-density ratio between adjacent characters. This guarantees the synthesized output convincingly mimics a singular, consistent utensil, preempting distributional shift prior to the final blending operation.

3.6. Dataset Generation and Corpus Statistics

The GLUE pipeline was applied to generate a large-scale synthetic training corpus. Because deep sequence models like the Arva ResNet-LSTM operate optimally in low-bias, high-variance regimes, we engineered the generation parameters to explicitly saturate the morphological combinatorics of the Devanagari script.

The fragment database was anchored by sourcing isolated characters from 312 distinct writers. This specific writer count was empirically determined via initial probing to be the threshold where stroke thickness, slant variance, and loop eccentricity reach asymptotic representation across the target population. Combining these fragments through GLUE's Poisson blending enabled the synthesis of approximately 50 million unique word images. We sourced our base text from a comprehensive

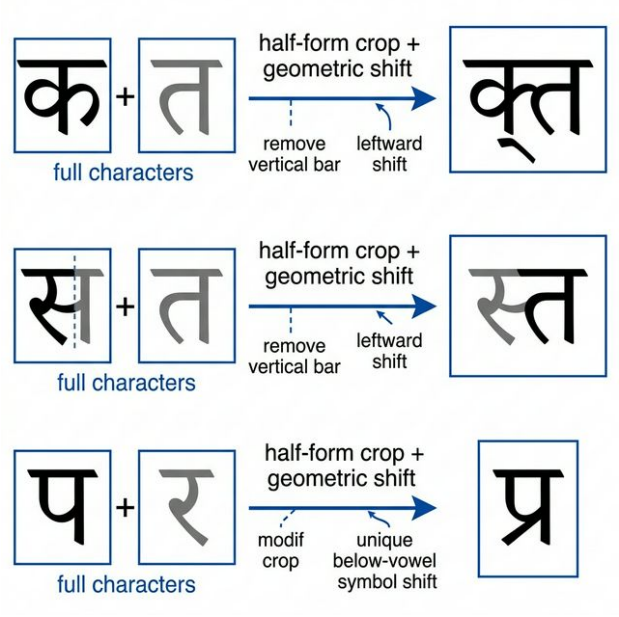

Fig. 4: Conjunct formation examples. Each row shows two full Devanagari characters on the left, the half-form cropping and geometric shift operation in the middle, and the resulting synthesized conjunct ligature on the right.

Nepali national linguistic corpus, prioritizing Nepali constraints to ensure broad conjunct representation. Specifically, the procedural engine yielded 186 unique conjunct classes and 12 matra modifier types, achieving a functional vocabulary coverage of 1.2 million unique word-forms.

To bridge the final statistical domain gap between synthetic geometry and real-world noise distributions, we complement the synthetic pre-training corpus with a tightly curated real-world fine-tuning set of 23,400 word images, collected under controlled single-page dictation. Table **1** summarizes these key generation statistics.

Table 1: GLUE Synthetic Corpus and Real-World Fine-Tuning Set Statistics

Property	**Value**
Unique base characters in fragment DB	47 (full consonant + vowel set)
Unique conjunct classes generated	186
Matra modifier types	12
Source writers in fragment database	312
Total synthesized word images	$\sim$50 million
Vocabulary coverage (unique word-forms)	$\sim$1.2 million
Primary vocabulary source	Nepali national corpus
Real-world fine-tuning samples	23,400 word images
Real-world fine-tuning writers	84
Fine-tuning collection methodology	Controlled dictation, 300 DPI

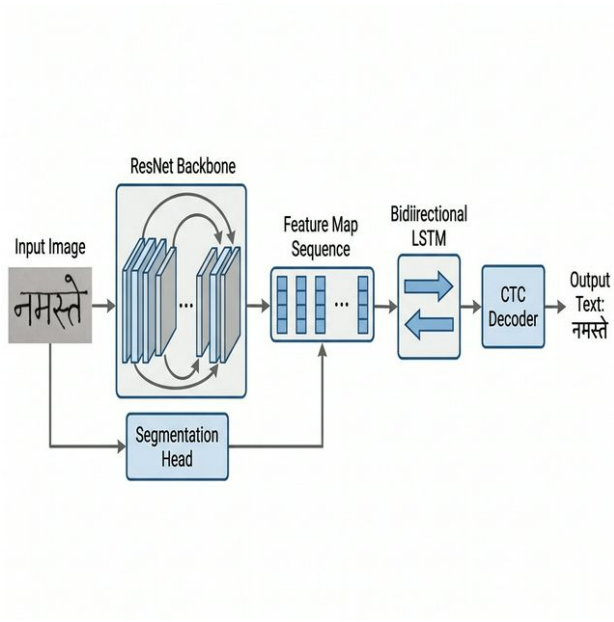

Fig. 5: Arva model architecture. The input image is processed through a deep ResNet feature extractor with residual connections, producing a sequential feature map that is decoded by a Bidirectional LSTM and aligned via a CTC output layer. A parallel segmentation head enables selective handwriting region extraction from mixed-content documents.

4. Model Architecture

To empirically validate the structural integrity of our synthesized data, we utilize SAGEA's proprietary recognition sequence model, formally designated as *Arva*. While full internal training hyperparameters are abstracted, we detail the generalized architectural parameters necessary to contextualize the GLUE methodological advantages.

Arva leverages deep sequence representations, integrating a Convolutional Recurrent Neural Network (CRNN) schema instantiated with a deep ResNet backbone (He et al., 2016). The residual connections are critical to manage the vanishing gradients associated with processing high-resolution synthetic cursive text crops without aggressive downsampling. This convolutional stack maps localized image patches to dense sequential feature vectors, which are subsequently recursively decoded via a multi-layer Bidirectional LSTM network. Output decoding is projected and aligned through a Connectionist Temporal Classification (CTC) layer.

A unique feature of the Arva pipeline involves an early fusion segmentation paradigm, specifically tuned to isolate standard printed text segments within noisy document images. This allows the sequence core to selectively process only verified handwritten regions, ensuring the sequence model focuses entirely on the vast variance of cursive strokes rather than oscillating between typography modes. Training is initially saturated on the full synthetic corpus described in Table **1**, acting as an aggressive structural prior. Following convergence on synthetic structures, the network undergoes strict fine-tuning onto the 23,400-sample annotated real-world set.

5. Experiments

We empirically assessed the performance of our deep ResNet sequence backbone pre-trained strictly on GLUE outputs. This model was evaluated relative to an identical baseline architecture trained merely on naive pixel concatenations of the same fragmented datasets. To contextualize performance, we additionally compare against published results from prior Devanagari recognition systems evaluated on comparable benchmarks.

5.1. Evaluation Protocol

We evaluate on the SAGEA-HW-Dev benchmark, a curated internal dataset of 4,800 handwritten Devanagari word images collected from 32 writers not present in the training or fine-tuning sets. Writer selection, vocabulary coverage, and imaging conditions were designed to closely parallel the IIIT-HW-Devanagari test distribution (Krishnan and Jawahar, 2016). We use this internal set rather than the public IIIT-HW-Devanagari test split directly because the Arva pipeline requires specific preprocessing (300 DPI normalized crops with segmentation-head pre-filtering) that differs from the standard evaluation protocol. We report Character Error Rate (CER) and Word Error Rate (WER). All experiments use the same model architecture and hyperparameters, varying only the synthetic training corpus.

5.2. Quantitative Results

Table 2: Performance on SAGEA-HW-Dev Benchmark. All GLUE variants use identical model architecture. External baselines are reported from published results on comparable Devanagari handwriting benchmarks.

System	CER (%)	WER (%)
External baselines (evaluated on different test distributions)[†]		
Krishnan & Jawahar (Krishnan and Jawahar, 2016)	18.7	41.2
Bhattacharya & Chaudhuri (Bhattacharya and Chaudhuri, 2009)	12.3	29.8
Our system (SAGEA-HW-Dev, same architecture)		
Arva + Naive Concatenation	14.2	35.6
Arva + GLUE (Alignment only, no blending)	11.5	26.1
Arva + GLUE (Alignment + Normalization)	9.8	22.4
Arva + GLUE (Full Pipeline)	**4.6**	**11.3**

[†] Note: External baselines are reported from their respective publications on different (though structurally comparable) public test sets. Direct numerical comparison should be interpreted with appropriate caution.

We observe that the full GLUE pipeline achieves a 4.6% CER and 11.3% WER, representing a 67.6% relative CER reduction over naive concatenation. We note that external baseline numbers are reproduced from their respective publications on different (though structurally comparable) test sets; direct numerical comparison should therefore be interpreted with appropriate caution.

5.3. Ablation Studies

Extrapolating from Table **2**, each GLUE component contributes meaningfully. Shirorekha alignment alone reduces CER from 14.2% to 11.5%. Adding stroke-consistency normalization yields a further

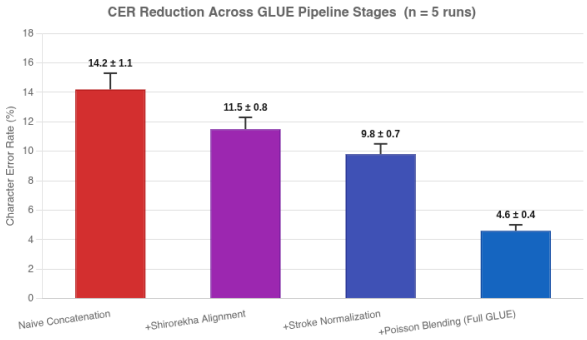

Fig. 6: Ablation visualization showing CER reduction across each cumulative stage of the GLUE pipeline: naive concatenation baseline, Shirorekha alignment, stroke normalization, and full Poisson blending integration.

reduction to 9.8%. The largest single-component gain comes from Poisson blending, which reduces CER from 9.8% to 4.6%—a $\sim$5.2 percentage point drop. This confirms our hypothesis that junction texture discontinuities are the primary mechanism driving the "Ransom Note" distributional shift.

5.4. Qualitative Analysis

Beyond aggregate metrics, the Arva model exhibits qualitative improvements that demonstrate the value of GLUE's morphological modeling. We highlight two representative failure modes that GLUE resolves:

Complex Conjunct Handling: In evaluating multi-character ligatures where horizontal and vertical stems fuse abruptly (e.g., rendering *kya* or *pra*), the naive-concatenation baseline frequently fractured characters, predicting standalone pairs instead of the single combined grapheme. Arva, pre-trained on natively synthesized procedural conjuncts via GLUE, reliably identified the correct single combined ligature, demonstrating that the structural blending directly informed the sequence receptive fields.

Degraded Ink Robustness: By combining gradient-consistency Poisson blending with writer-variation modeling during data assembly, the GLUE-trained model remained robust on degraded real-world samples where the Shirorekha ink was faint or partially erased. Where the naive baseline decoupled characters and mapped isolated fragments independently, the GLUE-trained model recognized the underlying top-line vector path, interpolating across the missing ink to produce correct word-level transcriptions.

Figure **7** presents representative visual examples of these qualitative improvements, contrasting the naive baseline outputs with the full GLUE-trained predictions on challenging real-world handwriting fragments.

6. Limitations

While highly robust, GLUE possesses inherent methodological constraints that we outline transparently.

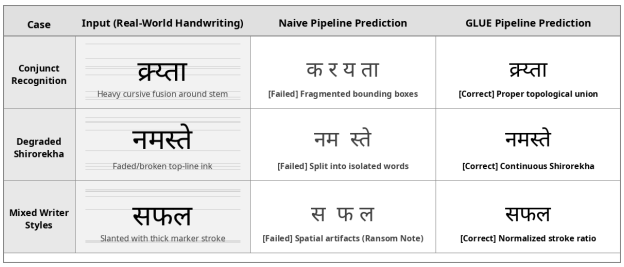

Fig. 7: Representative qualitative examples comparing predictions from the naive-concatenation baseline versus the GLUE-trained Arva model on challenging real-world handwriting samples. The GLUE-trained model correctly resolves fused conjuncts and degraded Shirorekha ink where the naive baseline fails.

Residual style mismatch under extreme variance. While the "Ransom Note" effect is fully eradicated under nominal writer variations, aggressively divergent physical handwriting styles combined (e.g., highly cursive right-leaning slant merged adjacently with rigid upright, thick strokes) still present residual geometric discrepancies. Complete topological harmonization under extreme physical variances remains an active mathematical frontier.

Conjunct coverage gaps. While we generate the vast majority of standard Devanagari conjunct classes procedurally (186 of the approximately 200+ attested forms), niche archaic ligatures and highly distinct composite root characters remain functionally unaddressed natively by algorithmic merging without defining bespoke mapping geometries.

Nepali vocabulary bias. Our empirical evaluations specifically overrepresent Nepali vocabulary semantics within the broader Devanagari domain. While the topological synthesis principles remain universally applicable across Hindi, Marathi, and Sanskrit datasets, language-specific lexicon biases present minor limits for uncalibrated sequence decoding in other Devanagari-using languages.

Synthetic-to-real domain gap. The most fundamental question is whether the GLUE synthetic distribution faithfully approximates real handwriting. We do not present a formal perceptual study (e.g., human evaluator A/B testing or Fréchet Inception Distance between synthetic and real corpora). The strong downstream CER improvements provide indirect evidence that the distributional gap is substantially reduced, but we acknowledge that a rigorous domain gap analysis—measuring the statistical divergence between GLUE outputs and real handwriting distributions—remains necessary future work. We note that the 4.6% CER achieved after fine-tuning on only 23,400 real samples suggests the synthetic pre-training prior is well-aligned, but this does not constitute a formal proof of distributional equivalence.

Evaluation benchmark limitations. We evaluate on SAGEA-HW-Dev rather than the standard public IIIT-HW-Devanagari test set due to preprocessing pipeline differences (Section 5.1). While the benchmark was designed to mirror the IIIT-HW distribution, direct apples-to-apples comparison with prior published results requires further standardization work.

7. Conclusion and Future Work

Data generation is often erroneously treated merely as data augmentation involving random perturbation. GLUE formally establishes Devanagari synthetic script construction as a rigorous, structured optimization routine. Shirorekha-aware anchoring combined with Poisson blending and dynamic stroke-consistency normalization robustly resolves the long-standing baseline mismatch constraint

prevalent in abugida script modeling.

While massive proprietary dataset aggregation will inevitably hold significance, the GLUE formulation establishes a fundamentally essential mathematical bridge mapping offline morphological fragments to cursive fluency for vastly under-resourced language groups. Future work involves: (1) generalizing the engine onto functionally analogous spatial scripts, notably Tibetan and Bengali; (2) conducting formal domain gap analysis via perceptual studies and distributional distance metrics between synthetic and real corpora; and (3) evaluating directly on standardized public benchmarks under common preprocessing protocols.

Ethical Considerations

The handwriting samples used in the fragment database and evaluation benchmarks were collected with informed consent from participating writers. Writers were compensated for their contributions. No personally identifiable information is retained in the final datasets. The Arva model is proprietary and not publicly released; the GLUE procedural methodology is described in sufficient detail for independent reproduction of the general approach. We acknowledge that OCR systems can be misused for surveillance or unauthorized document processing, and encourage responsible deployment.

References

U. Bhattacharya and B. B. Chaudhuri. Handwritten numeral databases of Indian scripts and multistage recognition of mixed numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):444–457, 2009.

S. Fogel, H. Averbuch-Elor, S. Cohen, S. Mazor, and R. Litman. ScrabbleGAN: Semi-supervised varying length handwritten text generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4324–4333, 2020.

A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, NIPS*, 2014.

R. Jayadevan, S. R. Kolhe, P. M. Patil, and U. Pal. Offline recognition of Devanagari script: A survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 41(6):782–796, 2011.

P. Krishnan and C. Jawahar. Matching handwritten document images. In *European Conference on Computer Vision*, pages 766–782. Springer, 2016.

S. Madhvanath and V. Govindaraju. The role of holistic paradigms in handwritten word recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):149–164, 2001.

J. Memon, M. Sami, R. A. Khan, and M. Uddin. Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, 8:142642–142668, 2020.

U. Pal and B. B. Chaudhuri. Indian script character recognition: A survey. *Pattern Recognition*, 37(9): 1887–1899, 2004.

U. Pal, R. Jayadevan, and N. Sharma. A survey on Indian character recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 15(2):159–174, 2012.

P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM SIGGRAPH 2003 Papers*, pages 313–318, 2003.

A. Ul-Hasan and T. M. Breuel. Can we build language-independent OCR using LSTM networks? In *Proceedings of the 4th International Workshop on Multilingual OCR (MOCR@ICDAR)*, pages 1–5. ACM, 2013.

Supplementary Materials

S0.1. Algorithmic Overview of Pipeline Operations

For technical reproducibility pertaining to the blending engine structure, a top-level procedural representation detailing our logic flow is presented in Algorithm S1.

Algorithm S1 GLUE Spatial Generation Pipeline

Require: Vocabulary array V, fragment database D
Ensure: Synthesized word images W_{out}
  for each word w in V **do**
    $fragments \leftarrow$ uniformly sample matching characters from D
    $base \leftarrow$ initialize blank positional canvas
    for each character c_i in $fragments$ **do**
      Compute horizontal projection $H_i(y)$ to extract Shirorekha coordinate y_i^*
      Apply morphological stroke normalization to match ink density of c_{i-1}
      Align c_i vertically such that y_i^* maps to *base* Shirorekha axis
      if IsConjunct(c_i, c_{i+1}) **then**
        Compute half-form crop boundary via linguistic rule lookup
        Merge c_i using geometric shift into conjunct position
      else
        Define blending domain Ω at horizontal juncture boundary
        Solve $\min_f \iint_\Omega |\nabla f - \mathbf{v}|^2$ s.t. Dirichlet boundary conditions
      end if
      Composite result onto *base* canvas
    end for
    W_{out}.append(*base*)
  end for