BengaliDiff: Diffusion Model for Few-Shot Bengali Font Generation

Md Bilayet Hossain[1,2], Honghui Yuan[1], Shabnur Anonna Akhy[1,2], and Keiji Yanai[1]

[1] Department of Informatics, The University of Electro-Communications, Tokyo 182-8585, Japan
[2] Department of Computer Science and Engineering, Daffodil International University, Dhaka 1216, Bangladesh
h2495009@gl.cc.uec.ac.jp, yuan-h@mm.inf.uec.ac.jp, shabnur15-3472@diu.edu.bd, yanai@cs.uec.ac.jp

Abstract. Bengali is a script-rich language with complex characters and ligatures, making it rare in the field of font generation. Existing font generation methods have achieved good results in Chinese, English, and other fonts. However, due to the complexity of the Bengali character, recent methods like FontDiffuser do not produce high-quality Bengali fonts. We propose BengaliDiff, a novel generative model that uses a diffusion-based architecture, style-content fusion, and adversarial supervision to synthesize Bengali characters in a target font style. We use image-to-image translation-based methodology, which enhances font production, as we maintain the structure of characters and provide them with a uniform style in different fonts. We build on our approach of FontDiffuser but use a dual aggregation cross-attention scheme to inject content and style features on channel and spatial levels, individually, into the reverse denoising process. In addition, we embed an adversarial discriminator that promotes stylistically coherent and perceptually accurate generations. According to tests performed with a predefined group of Bengali fonts, it can be said that BengaliDiff is better in content preservation and style consistency compared to the current baselines that exist. To the best of our knowledge, our method is the first to use the diffusion model for Bengali font generation task. The study also provides a publicly available Bengali font dataset and a pre-trained model that allows them to support digitally published materials, text handwriting recognition, and custom typography with better assistance.

Keywords: Font Generation, Diffusion model, Bengali Font

1 Introduction

Bengali is ranked among the popular languages of the world and is spoken by more than 200 million people. It has a rich and distinctive text that constitutes a significant component of the culture and heritage. Meanwhile, Bengali fonts of good quality are very limited when compared to Latin fonts or other

scripts. Building new Bengali fonts by hand requires skilled designers and significant time, as the letters of the Bengali language are of complex shapes. This stimulates the necessity of automatic generation of the Bengali fonts, which may save time on the work and encourage the design of new fonts. One of the biggest challenges in digital typography for a long time has been producing visually coherent and stylistically consistent fonts from a small number of references [2,9,18]. There are unique issues in Bengali script with its multi-glyph structure and diverse glyph composition, which are not addressed by generic font producing methods developed to support Latin or logographic languages (like Chinese). The robust model of Bengali font synthesis must address a number of underlying problems, such as conjunct forms and complex ligatures, matra (rendering and baseline positioning of floating vowel signs). From style transfer to GAN-based font generation [16], traditional generative approaches [10,7,14] have trouble with these script-specific subtleties. Latin and Chinese letters have received interesting results under a noise-to-noise model recently introduced to diffusion models, such as FontDiffuser [19] and Diff-Font [6]. However, when these models are directly applied to Bengali without explicitly describing their structural features, they frequently have poor generalization. In this paper, we propose BengaliDiff, a diffusion-based font generation framework designed to operate in a one-shot environment. BengaliDiff has presented new training and architectural improvements.

- Adversarial supervision discriminator: The diffusion models are not pixel-level realistic on the stroke level; however, they generate realistic images by iteratively refining them. To address this, we add a patch-level discriminator that discourages artifacts and encourages sharper results.
- Cross-Attention Content Fusion: At several levels within the U-Net architecture, our transformer-based cross-attention combines the style and content elements. This makes it easier for the model to synthesize glyphs that retain their intricate structure while also adopting the desired style.

Our contributions are summarized as follows:

1. We proposed a unique Bengali script-specific diffusion model.
2. We introduced a new joint hybrid training goal involving diffusion denoising, adversarial learning, and contrastive style guidance.
3. We achieved state-of-the-art performance on Bengali few-shot font generation benchmarks.

This is how the rest of the paper is organized. Related work is reviewed in Section 2. Our training technique and model architecture are shown in Section 3. The experimental design, as well as the quantitative and qualitative findings, are described in Section 4. The broad discussion and future directions are presented in Section 5. The paper is finally concluded in Section 6.

2 Related Work

To the best of our knowledge, there is currently no literature on Bengali font generation using a diffusion approach. It should be mentioned that studies on the use of GANs for the generation of handwritten Bengali characters have been conducted. This section provides an overview of the related studies of this work, BengaliDiff.

2.1 Font Generation

Previous approaches in font generation [3] applied direct style-content disentanglement like the use of VAE-GANs approaches [1] or supervised component-based GANs [7]. Representative works like LF-Font [10] used a factorization strategy, and CG-GAN [7] decomposed glyphs into shared primitives or strokes, which were used to transfer styles between glyph sets [2]. Diff-Font [6], a one-shot font generation framework based on a multi-attribute conditional diffusion model. Unlike GAN-based methods, Diff-Font achieves stable training and high-quality generation, especially for complex glyphs in Chinese and Korean. This method is hindered by low inference speed, unreliable recognition of rare characters, and poor generalization to unseen glyphs due to token dependence.

Moreover, MSD-Font [4] proposed a multi-stage few-shot font generation model built on latent diffusion, which is inspired by expert designers' workflow. This model separates the generative process into structure construction, font transfer, and refinement stages. They have Limitations, including higher inference time and model size, due to the complexity of diffusion models. DG-Font [17] presented an unsupervised font generation model leveraging deformable convolution and a novel Feature Deformation Skip Connection (FDSC) to capture geometric style variations. Additionally, a number of approaches have been proposed to expand the above work. A self-supervised cross-modality pre-training method was presented by XMPFont [8]. MX-Font++ [15] introduced an enhanced few-shot font generation model that uses Heterogeneous Aggregation Experts (HAE) for improved feature extraction. This method contributes a novel content-style homogeneity loss to better disentangle content and style in the latent space. Early works like Auto-Encoder Guided GAN [9] and MC-GAN [2] applied encoder-decoder and GAN-based architectures to generate Chinese or Latin fonts using multiple reference images. Their model is effective for simpler scripts, and these models require a large number of examples and often struggle with structural consistency in more complex scripts. However, based on our study, most of these methods did not work for generating Bengali fonts with complex structures, such as characters with more looped and curved stroke patterns.

2.2 Diffusion Model Methods

In recent years, Denoising Diffusion Probabilistic Models (DDPMs) [5] have achieved state-of-the-art results in image synthesis. Some methods have been pre-

sented, such as Okkhor-Diffusion [5], a novel framework for class-guided generation of Bangla isolated handwritten characters using a DDPM. The model outperformed StyleGAN2-ADA in visual and structural quality on multiple datasets, including BanglaLekha-Isolated. It achieved strong results with FID, MS-SSIM, LPIPS, and introduced a new metric, BCAFID, specifically designed for Bangla evaluation. Similarly, VecFusion [13] employed a two-stage cascaded diffusion model for generating editable vector fonts from glyph codepoints and font style inputs. A novel discrete-continuous representation enables precise control point prediction across diverse glyph structures. In the context of font generation, FontDiffuser [19] applied a conditional diffusion model with multi-scale content fusion and contrastive style supervision for Chinese fonts. Their method effectively handles complex characters and large style variations. Experiments on multiple benchmarks show that FontDiffuser outperforms state-of-the-art GAN-based methods in quality and generalization. As our method also utilizes the diffusion model, BengaliDiff expands on this concept by adding new architectural elements that are appropriate for Bengali scripts.

2.3 Indic Scripts and Bengali Typography

Bengali language representation is extremely difficult: the same basic character can be radically different in its visual forms depending on either the lack or manner of application of a diacritic mark or conjunct. Few prior works have addressed Bengali font generation beyond handwriting or optical character recognition (OCR). Some methods, like the VAE-GAN-based [1] model, for generating Bangla printed characters from handwritten inputs to support OCR and preserve rare compound characters. Using the CMATERdb [12] dataset and a printed font set, their architecture with a U-Net-based generator and CNN-based discriminator learns image-to-image translation effectively. Their Future work includes improving image quality, increasing model depth, and applying the system to full document digitization. Moreover, BBCNet-15 [11] introduced a deep convolutional neural network for Bangla handwritten basic character recognition. The model, consisting of 15 layers including convolutional, pooling, and fully connected layers with dropout regularization, was trained on the CMATERdb 3.1.2 dataset. It achieved a test accuracy of 96.40%, outperforming the previous methods, including SVM and earlier CNN models. Our approach varies from the previous approaches in that it concentrates on contemporary typefaces, even if it also uses the diffusion model and reference font images for Bengali font development.

3 Methodology

3.1 Proposed Method Overview

Our proposed BengaliDiff method is a conditional diffusion model-based image generation method. The input of our method is a reference style glyph and a

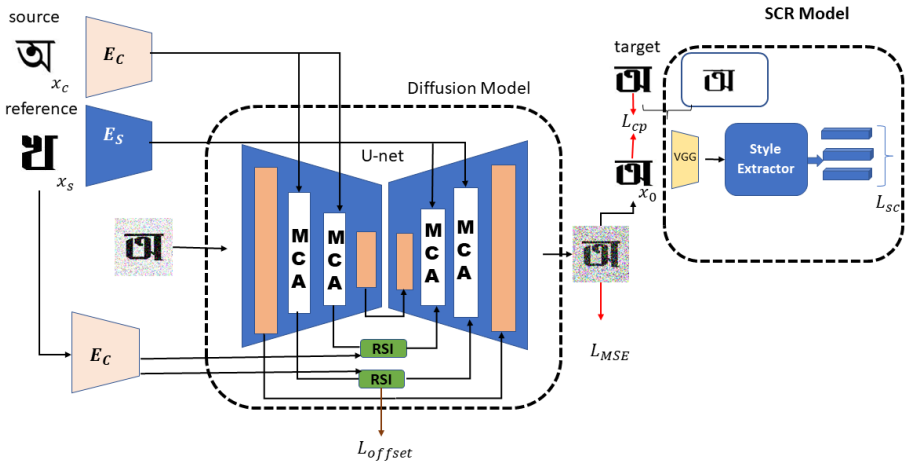

Fig. 1. The framework of the base method FontDiffuser.

source content glyph. Then, the model creates a new glyph that imitates the style while keeping the original content. Recently, many methods for font generation have achieved significant results. FontDiffuser [19] has shown impressive results on various fonts. It has created high-end results, especially when dealing with complex fonts and styles, which involve more serious adjustments as compared to earlier methods. We used FontDiffuser as our base network for generating Bengali fonts. Fig. 1 shows the overall framework of the base method, FontDiffuser. This architecture includes a content encoder E_c to extract structural features from a source glyph x_c, a style encoder E_s to represent font style x_s, and a conditional U-Net-based diffusion model that predicts clean glyphs by gradually denoising random noise. In addition, the U-Net has two additional modules, the Multi-Scale Content Aggregation (MCA) module injects multi-resolution content features that keep structural detail, and the Reference-Structure Interaction (RSI) module, which employs the deformable convolution to align the spatial features between reference and source glyphs. FontDiffuser has a two-stage training policy that leads to gradually learning to reconstruct the correct structures and simulate style uniformity. In the first phase, only the diffusion model is trained without the style contrastive refinement (SCR) module. The process is devoted to reconstructing glyphs based on feature and structural losses. The total loss is defined as follows:

$$\mathcal{L}^1_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda^1_{\text{cp}}\mathcal{L}_{\text{cp}} + \lambda^1_{\text{off}}\mathcal{L}_{\text{offset}} \tag{1}$$

$$\mathcal{L}_{\text{MSE}} = \|\epsilon - \epsilon_\theta(x_t, t, x_c, x_s)\|^2 \tag{2}$$

$$\mathcal{L}_{\text{cp}} = \sum_{l=1}^{L} \|\text{VGG}_l(x_0) - \text{VGG}_l(x_{\text{target}})\| \tag{3}$$

$$\mathcal{L}_{\text{offset}} = \text{mean}\left(\|\delta_{\text{offset}}\|\right) \tag{4}$$

Three sub-components are combined with the loss function. A Mean Square Error (MSE) loss, a content perceptual loss and a deformation offset loss. Specifically, MSE was used to measure the difference between predicted noise ϵ_θ and true noise ϵ. The loss function encourages the model to accurately predict noise in back-diffusion and generate images that recover the original image. Content perceptual loss using the VGG network to measure the degree of similarity between the generated image x_0 and the target image x_{target} in terms of deep semantic features. The deformation offset loss uses deformable convolutional networks (DCN) to constrain the offset of content features δ_{offset}. This is used to prevent the network from generating excessive offsets or unstable behavior during the generation process.

In Phase 2, the SCR module is switched on to direct the model to learn style imitation at the global and local levels. Style features of every style image are obtained using a style extractor. This phase introduces a new style contrastive loss term (L_{sc}), which enforces similarity among glyphs from the same font style and separation from other styles. The updated loss becomes:

$$\mathcal{L}^2_{\text{total}} = \mathcal{L}_{\text{MSE}} + \lambda^2_{\text{cp}}\mathcal{L}_{\text{cp}} + \lambda^2_{\text{off}}\mathcal{L}_{\text{offset}} + \lambda^2_{\text{sc}}\mathcal{L}_{\text{sc}} \tag{5}$$

Together, the two-phase training approach enables FontDiffuser to produce coherent and stylistically rich font sets, making it an effective base for further adaptation to complex scripts such as Bengali. However, unlike Chinese and English, Bengali text has a special structure, such as splicing, which makes it difficult for existing methods to generate clear text. Fig. 2 shows the result of generating Bengali fonts using FontDiffuser. The result shows that FontDiffuser is not able to generate the Bengali text correctly, there are large distortions in the font, and it is also fails to apply the Stylistic features to the text.

Fig. 2. The results generated by FontDiffuser using Bengali font as the input image.

3.2 Cross-Attention Content Fusion

As shown in Fig. 3, On the U-Net architecture, in our model of Bengali font generation, we enhance the building block by introducing a dual aggregation Cross-Attention Content Fusion (CACF) module to the network at every level of the encoder and the decoder network. This module assists the model to improve integration of content of a source character with style of a reference font. Under this scheme, the source glyph content features are employed as queries and style features of the reference glyph are employed as keys and values. This enables the model to use the most beneficial details of style in the production of a new glyph. Our dual aggregation cross-attention is intended to concentrate on both the features (channel information) and the location of objects in the image (spatial information). The model can better comprehend subtle brush patterns, curves, and the overall look of the style by digesting these two kinds of data. The attention system selects the most crucial style elements and gradually incorporates them into the content characteristics.

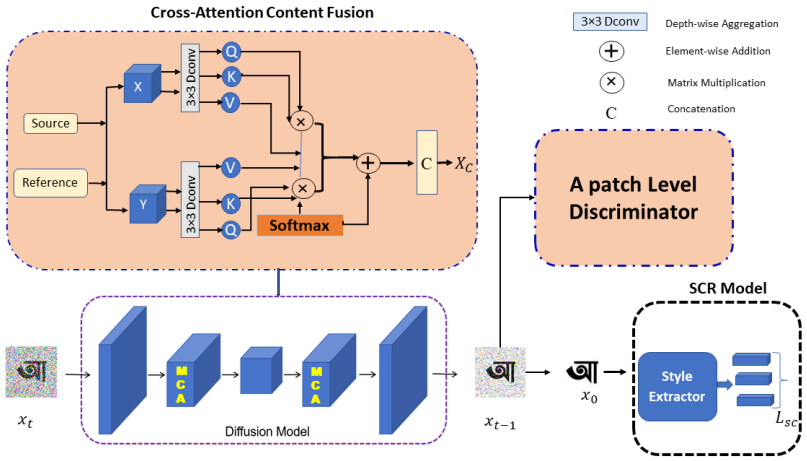

Fig. 3. Overview of our proposed method. The Cross-Attention Content Fusion and A Patch Level Discriminator in the blue dashed line boxes.

By placing this attention mechanism in the downblock and upblock of the U-Net of the diffusion model ensures that the style is used everywhere with the whole network. This gives a better and cleaner character in the model particularly composite Bangla characters with matras (horizontal shapes), loops and compound characters. The model is trained how to pair style with content in a better way, thus being able, even after a few examples, to create high-quality font images.

Overall, this dual aggregation cross-attention content fusion module plays a key role in making our model better at transferring style and producing Bengali fonts that are both visually appealing and structurally correct.

3.3 Discriminator for Adversarial Supervision

We add a discriminator at patch level which is based on a CNN and our purpose of this discriminator is to provide a direction in which the model is to be trained. Our discriminator is to validate that glyph generated is either real or fake. We crop tiny pieces (patches) of the picture and say whether they are produced on the basis of the real fonts or developed. The discriminator and generator (our diffusion model) are trained in adversarial fashion. The generator attempts to produce glyphs that will dupe the discriminator and the discriminator attempts to pick up such fakes. Setting Our layout assists the generator to produce sharper and more realistic glyphs particularly in places where details are fine such as thin strokes, loops, or ornamentation ends. Including a discriminator enhances the final output clarity and sharpness as opposed to when only diffusion loss was used during training. FontDiffuser training method only takes advantage of the diffusion loss to better fine-tune the generated glyphs through numerous iterations. Sometimes, it generates less sharp images, even though the structure works well. However, our BengaliDiff Model extends it with a patch-based discriminator, that is conditioned to distinguish small elements (patches) of a glyph as real or generated. The discriminator supplies the model with extra data on the texture of a font and the clarity of the strokes. Specially improves the sharpness and fine-detail quality of generated images, particularly in difficult places like matras, curving tails, and complex ligatures. This improvement is clearly shown in both visual comparisons and our quantitative data (LPIPS and FID). In summary, the patch-level discriminator acts as a powerful corrective mechanism, guiding the BengaliDiff generator toward producing glyphs that are not only structurally accurate but also visually pleasing and typographically robust, far surpassing the outputs of diffusion-only baselines.

4 Experiments and Results

4.1 Datasets and Evaluation Metrics

We use Kaggle to collect 55 Bengali fonts (styles). For the training set, we choose 50 fonts at random (as “seen fonts”), each of which has 800 unique characters. We employ two tests to assess the methods: one testing set comprises 5 unseen fonts with 162 unseen characters (known as “UFUC”) and 10 seen fonts with 120 unseen characters that are not seen during training (known as “SFUC”). For quantitative analysis, we employ the FID, SSIM, LPIPS, RMSE, and L1 loss measures.

4.2 Implementation Details

We train FontDiffuser with $\beta 1 = 0.9$ and $\beta 2 = 0.999$ using the AdamW optimizer. The image's dimensions are set at 96×96. In phase 1, we train the model using a batch size of 16 and a total of 90000 steps. The learning rate with a linear schedule is 1e−4. The learning rate is set as 1e−5 for the second phase. For training, we utilize 16 negative samples, 30000 steps total, and a batch size of 16. One RTX 3090 GPU is used for both training and testing.

4.3 Comparison with State-of-the-Art Method

We compare our approach with the baseline method, FontDiffuser. It designed with the similar collection of Bengali letters. Fig. 4 contains the detailed visual example of the FontDiffuser results, the BengaliDiff model, and the target glyphs. This research demonstrates that BengaliDiff has certain advantages in several important font generation domains. Though it operates effectively in preserving overall shape, FontDiffuser is often challenged with smaller visual elements of the Bengali script. As an example, the font FontDiffuser is rendered with bad loops, interruptions of the stroke, and even dysfunctional matra positions within the characters. The glyphs can also sometimes be clipped or structurally degraded because Fontdiffuser has not been carefully monitored, so it is either blurry or the stroke thickness is not consistent, especially when it has a design that is curved or running diagonally. This issue is more observable in complicated conjuncts and heavy ligature characters.

BengaliDiff, however, outputs glyphs that are structurally and stylistically rather close to the ground truth. The discriminator fixes the distortions that often appear in the outputs of FontDiffuser and guides the model to generate more precise and clean stroke edges. This creates more visually clean characters, as may be seen in our version of complex characters, which quite obviously does not remove all minor decoration such as loops and hooks. As well, BengaliDiff can effectively identify specific areas of content, that is, simple consonants and matras, with their respective equivalent stylistic components of the reference as a result of the cross-attention approach. This is most so in glyphs, where matras are smoothly integrated into the character structure with the ratio of thickness, curve, and alignment. BengaliDiff accurately captures both local texture and global layout with the inclusion of multi-scale content characteristics and style-aware attention.

A key advantage of BengaliDiff is that it is consistent within the set size, stroke thickness are same, sigma rhythm of the strokes are same which is in contrast to fontdiffuser which original and generated strokes were not always the same, but greatly differed in stroke thickness or character shape interpretation no matter how close the two forms were. This uniformity is very important in terms of professional font design and typesetting in Bengali since it preserves intelligibility and aesthetic harmony. Despite all of the above, through the visual productions on Fig. 4, one can say that BengaliDiff not only overcomes the

Fig. 4. Qualitative Results Comparisons on SFUC between our method and the previous state-of-the-art method of FontDiffuser. The red boxes highlight the challenging areas of FontDiffuser.

weaknesses of FontDiffuser: BengaliDiff presents characters that are not only aesthetically appealing, structurally consistent and typographically correct. These additions justifies why the discriminator and cross-attention are important elements of our model. To perform a quantitative analysis of our method with FontDiffuser, we used five commonly adopted image quality metrics: FID, LPIPS, L1, RMSE, and SSIM. FID (Fréchet Inception Distance) measures the difference in distribution between generated and real images, providing a strong indicator of overall visual realism. LPIPS evaluates perceptual similarity based on deep features, capturing differences that matter to human perception. L1 and RMSE compute pixel-level errors between the generated and ground truth images, assessing low-level fidelity. SSIM measures structural similarity, emphasizing the preservation of local textures and shapes.

Table 1. Quantitative evaluation results on SFUC with FontDiffuser methods.

Model	FID↓	LPIPS↓	L1↓	RMSE↓	SSIM↑
FontDiffuser	0.8685	0.3568	**0.2377**	**0.3151**	**0.7131**
Ours	**0.7063**	**0.3223**	0.2547	0.3240	0.6751

Fig. 5. Qualitative Results Comparisons on UFUC between our method and the previous state-of-the-art method of FontDiffuser. The red boxes highlight the challenging areas of fontDiffuser.

Table 2. Quantitative evaluation results on UFUC with FontDiffuser methods.

Model	FID↓	LPIPS↓	L1↓	RMSE↓	SSIM↑
FontDiffuser	0.9573	0.3738	**0.2419**	**0.3142**	**0.6855**
Ours	**0.7280**	**0.3406**	0.2706	0.3357	0.6491

We conducted experiments on a set of 100 characters where the characters vary in the nature of Bangla, such as simple and compound Bangla characters, as shown in Table 1. Our method achieved significantly better results in FID and LPIPS, with improvements of 0.1622 and 0.0345, respectively, better than FontDiffuser. This shows that our model not only provides more realistic images, but the differences with regard to the target fonts are also perceptually closer. Our approach yielded a higher score in L1 and RMSE, albeit slightly, but the scores are close enough that it implies a slight trade-off of pixel-level accuracy. The structural similarity is also preserved well since our SSIM scores are also fairly close to each other. Table 2 demonstrates that for unseen fonts and characters, BengaliDiff is more effective than the FontDiffuser. It has better results that are more realistic and visually closer (lower FID and LPIPS) and maintains the structure properly. The L1 and RMSE small differences indicate that trade-offs are only minor at the pixel level and that the SSIM should also have excellent

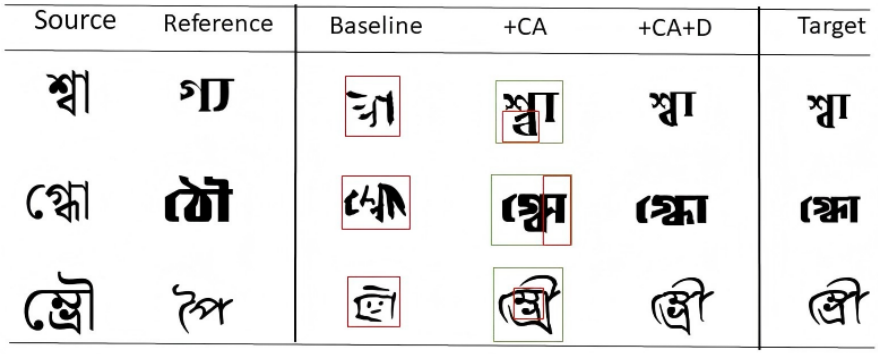

Fig. 6. Qualitative evaluation results of ablation studies. An illustration of several modules. CA and D represent Cross-attention and Discriminator, respectively. Red boxes represent the missing strokes, while green represents the corresponding improvements.

shape preservation. Fig. 5 visually compares both methods on UFUC. BengaliDiff generates characters with correct shapes but sometimes fails to match the style of the reference font, like stroke thickness or curve design. FontDiffuser, in contrast, often shows distorted or broken glyphs. Overall, BengaliDiff is better at preserving content, but style transfer to unseen fonts still needs improvement.

All these results showed that our model produces better-quality font images on average and has a good generalization capability to a wide variety of Bangla characters and produces font images of a better aesthetic quality comparatively, making it more useful in real-world OCR applications and digital typography.

4.4 Ablation Studies

We conducted ablation tests to evaluate the efficacy of every component of our approach. our qualitative results presented in Fig. 6 illustrate effectiveness of various elements in our model via ablation studies, and Table 3 shows the Quantitative evaluation results of ablation studies. Specifically, Fig. 6 compares the visual quality of generated Bangla characters across several model configurations: Baseline, Baseline + Cross-Attention (CA), and Baseline + Cross-Attention with Discriminator (CA + D), alongside the ground truth (Target) and reference font. The example of the input character and the style can be seen in the source and the reference columns. Looking at the Baseline column, we can see how the overall shape of characters still remained intact, but we can notice a significant number of distortions in the complicated strokes and misrepresentation of ligatures. This indicates that the baseline does not have the capability to capture any form of detailed style information.

When we add the Cross-Attention (CA) module, the visual results improve significantly. The CA module allows better alignment between the reference style and the target character structure, leading to clearer glyph shapes and more accurate stroke positioning. For example, in the second row, the complex character

Table 3. Quantitative evaluation results of ablation studies. Effectiveness of different modules. CA and D represent Cross-attention and Discriminator, respectively. The first row represents the Baseline.

	FID↓	LPIPS ↓	L1 ↓	RMSE↓	SSIM ↑
baseline	0.5554	0.2918	0.1579	0.2430	0.7680
CA	0.2372	0.1647	0.1613	0.2432	0.7714
CA & D	**0.0877**	**0.0909**	**0.1031**	**0.1768**	**0.8345**

shows better integration of the matra (horizontal stroke) and conjunct formation compared to the baseline. However, although CA improves style transfer, some inconsistencies remain in the finer details, particularly in maintaining the consistency of line thickness and stroke endpoints.

The full model, which includes both cross-attention and a discriminator (CA & D), produces the most visually pleasing and structurally accurate results. With the discriminator, the model gets to learn how to produce outputs that not only adhere to the reference style, but are also incomparable to real font samples, in overall distribution. This is evident in the final column before the target, where the generated images closely resemble the target fonts in terms of shape, stroke consistency, and spatial arrangement. The characters appear more balanced, smooth, and clean compared to the previous versions.

Overall, this ablation study clearly highlights the contributions of each component in our model. The cross-attention mechanism is vital in transferring stylistic properties of a reference whereas the discriminator makes it realistic and consistent. These results show that our proposed modules significantly improve the ability to generate high-quality Bangla fonts, especially for complex characters involving ligatures and diacritics. This analysis confirms that both elements are necessary in creation of outputs which are structurally correct and stylistically true to the reference font.

5 Discussion

Our results justify the value of each module that we presented. Cross-attention enhances fine stroke transfers, whereas the discriminator enhances sharpness and fine detail of generated Bengali fonts. But our model is actually trained on the persistently executed fonts, which are digital, clean, uniform, and adhere to the typographic rules. The model used by us is based on rendered fonts, and this reduces the reference to handwriting or artistic scripts. It makes a fixed set of templates of ligatures as well. Control of style is poor, and the model demands a lot of computational resources. We will provide handwritten glyphs in the future, allow dynamic ligature generation, and create style vectors, which are editable by the user. While we report perceptual and structural metrics like FID, LPIPS, and SSIM, we acknowledge that OCR-based evaluation (e.g., CER) would better reflect real-world usability. Due to time and resource constraints, we could not include such evaluation in this version and also compared only with FontDiffuser,

the most relevant diffusion-based baseline for evaluating our improvements. We plan to explore CER-based benchmarking in future work for deeper task-specific assessment.

Fig. 5 demonstrates BengaliDiff has issues during the generation of Unseen font with unseen characters (UFUC). Although the model produces the content of the character correctly, most of the time it fails to resemble the style of the font being referenced. What this implies is that our model is retaining the right letter and altering the font style, which is not the objective. As an example, the strokes, shapes, and general appearance of the characters are not perfectly aligned with the reference font. To enhance BengaliDiff, it may also be possible to train on more diverse fonts (handwritten or artistic fonts) to enhance generalization. Besides, adding more flexible style modeling (dynamic ligature support or style vectors that can be edited by the user) could allow the model to more easily capture reference styles. The design of lighter-weight architectures or explicit style-contrast mechanisms represents future research directions to further improve the performance of unseen fonts.

6 Conclusion

In this paper, we present a novel framework that involves diffusion-based generation of Bengali font using a dual aggregation cross-attention and a patch-level CNN-based learning discriminator module. To the best of our research, no previous studies had applied the approach of the diffusion to the generation of Bengali fonts. Based on our experimental results, our proposed method effectively generates Bengali fonts while maintaining their structural soundness. The implementation of our proposed dual aggregation cross-attention has been done successfully in terms of combining style information on the reference glyph with the content of the source glyph. Since our proposed method works on the digitally rendered fonts, we plan to expand our model in the future to accommodate handwritten glyphs and to improve the performance of unseen fonts.

Acknowledgments: This work was supported by JSPS KAKENHI Grant Number, 22H00548, and JST CRONOS Grant Number, JPMJCS24K4.

References

1. Abedin, M.M.h.z., Ghosh, T., Mehrub, T., Yousuf, M.A.: Bangla printed character generation from handwritten character using gan. In: Soft computing for data analytics, classification model, and control, pp. 153–165. Springer (2022)
2. Azadi, S., Fisher, M., Kim, V.G., Wang, Z., Shechtman, E., Darrell, T.: Multi-content gan for few-shot font style transfer. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7564–7573 (2018)
3. Chang, J., Gu, Y., Zhang, Y., Wang, Y.F., Innovation, C.: Chinese handwriting imitation with hierarchical generative adversarial network. In: Proc. of the British Machine Vision Conference. p. 290 (2018)

4. Fu, B., Yu, F., Liu, A., Wang, Z., Wen, J., He, J., Qiao, Y.: Generate like experts: Multi-stage font generation by incorporating font transfer process into diffusion models. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 6892–6901 (2024)
5. Fuad, M.M., Faiyaz, A., Arnob, N.M.K., Mridha, M.F., Saha, A.K., Aung, Z.: Okkhor-diffusion: class guided generation of bangla isolated handwritten characters using denoising diffusion probabilistic model (ddpm). IEEE Access (2024)
6. He, H., Chen, X., Wang, C., Liu, J., Du, B., Tao, D., Yu, Q.: Diff-font: Diffusion model for robust one-shot font generation. International Journal of Computer Vision **132**(11), 5372–5386 (2024)
7. Kong, Y., Luo, C., Ma, W., Zhu, Q., Zhu, S., Yuan, N., Jin, L.: Look closer to supervise better: One-shot font generation via component-based discriminator. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 13482–13491 (2022)
8. Liu, W., Liu, F., Ding, F., He, Q., Yi, Z.: Xmp-font: Self-supervised cross-modality pre-training for few-shot font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7905–7914 (2022)
9. Lyu, P., Bai, X., Yao, C., Zhu, Z., Huang, T., Liu, W.: Auto-encoder guided gan for chinese calligraphy synthesis. In: 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR). vol. 1, pp. 1095–1100. IEEE (2017)
10. Park, S., Chun, S., Cha, J., Lee, B., Shim, H.: Few-shot font generation with localized style representations and factorization. In: Proc. of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2393–2402 (2021)
11. Saha, C., Faisal, R.H., Rahman, M.M.: Bangla handwritten basic character recognition using deep convolutional neural network. In: 2019 Joint 8th International Conference on Informatics, Electronics & Vision (ICIEV) and 2019 3rd International Conference on Imaging, Vision & Pattern Recognition (icIVPR). pp. 190–195. IEEE (2019)
12. Sarkar, R., Das, N., Basu, S., Kundu, M., Nasipuri, M., Basu, D.K.: Cmaterdb1: a database of unconstrained handwritten bangla and bangla–english mixed script document image. International Journal on Document Analysis and Recognition (IJDAR) **15**, 71–83 (2012)
13. Thamizharasan, V., Liu, D., Agarwal, S., Fisher, M., Gharbi, M., Wang, O., Jacobson, A., Kalogerakis, E.: Vecfusion: Vector font generation with diffusion. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 7943–7952 (2024)
14. Wang, C., Zhou, M., Ge, T., Jiang, Y., Bao, H., Xu, W.: Cf-font: Content fusion for few-shot font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 1858–1867 (2023)
15. Wang, W., Sun, D., Zhang, J., Gao, L.: Mx-font++: Mixture of heterogeneous aggregation experts for few-shot font generation. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 1–5. IEEE (2025)
16. Wang, X., Li, C., Sun, Z., Hui, L.: Review of gan-based research on chinese character font generation. Chinese Journal of Electronics **33**(3), 584–600 (2024)
17. Xie, Y., Chen, X., Sun, L., Lu, Y.: Dg-font: Deformable generative networks for unsupervised font generation. In: Proc. of IEEE Computer Vision and Pattern Recognition. pp. 5130–5140 (2021)
18. Xie, Y., Chen, X., Zhan, H., Shivakumara, P., Yin, B., Liu, C., Lu, Y.: Weakly supervised scene text generation for low-resource languages. Expert Systems with Applications **237**, 121622 (2024)
19. Yang, Z., Peng, D., Kong, Y., Zhang, Y., Yao, C., Jin, L.: Fontdiffuser: One-shot font generation via denoising diffusion with multi-scale content aggregation and

style contrastive learning. In: Proc. of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 6603–6611 (2024)