DiaFiLM: Representation-Modulated Dialect-Aware Grapheme-to-Phoneme Conversion for Bengali

Anonymous ACL submission

Abstract

Modeling dialectal pronunciation variation remains a central challenge for grapheme-to-phoneme (G2P) conversion in low-resource languages such as Bengali, where phonological realizations differ systematically across geographic regions. Existing neural transcription systems typically assume a standardized language form and incorporate dialect information only as symbolic input, limiting their ability to capture region-specific phonotactic patterns. In this work, we propose **DiaFiLM**, a lightweight (12.5M parameter) dialect-aware sequence-to-sequence architecture that models regional pronunciation differences through feature-wise modulation of internal representations. The proposed model integrates phonetic feature-informed grapheme encoding and local phonotactic modeling to improve robustness in dialectally diverse settings. Experiments on a multi-district Bengali IPA corpus demonstrate that DiaFiLM consistently outperforms strong neural and rule-based baselines, achieving a macro-average Word Error Rate of 0.296 and Character Error Rate of 0.088. The results highlight the effectiveness of explicit dialect-conditioned representation learning for grapheme-to-phoneme conversion and suggest a general strategy for improving pronunciation modeling in multilingual and multi-dialect low-resource scenarios. DiaFiLM is publicly available at https://anonymous.4open.science/r/DiaFiLM-Team/.

1 Introduction

Bangla is spoken by more than 300 million people worldwide and exhibits substantial linguistic variation across its regional dialects. These dialects differ not only lexically but also in phonology, morphology, and syntax, leading to systematic differences in pronunciation across geographic regions. Modeling such variation remains a fundamental challenge for computational language technologies that rely on consistent phonetic representations. In particular, grapheme-to-phoneme (G2P) conversion, or transcription of Bangla text into the International Phonetic Alphabet (IPA), becomes considerably more complex when dialectal variation is taken into account.

A central research question therefore arises: how can computational models effectively capture dialect-specific pronunciation patterns while maintaining robustness across multiple regional variants of a language? This question is especially important for Bangla, where phonological realizations vary significantly between districts, yet dialect-specific annotated resources remain limited. Traditional text-to-IPA systems are typically trained on standardized Bangla corpora and often fail to generalize to regional pronunciations due to phonetic variability and insufficient dialect-aware supervision. Furthermore, context-dependent phenomena such as numeral verbalization, heteronym disambiguation, suffix-induced vowel lengthening, and loanword adaptation introduce additional ambiguity in phonetic transcription (Fatema et al., 2024a; Islam et al., 2024a). Robust dialect-aware IPA transcription is therefore essential for downstream applications including Automatic Speech Recognition (ASR), Text-to-Speech (TTS) synthesis, linguistic documentation, and language learning technologies, particularly in low-resource dialect settings.

Recent research has begun to address Bangla IPA transcription using neural approaches. The DUAL-IPA dataset provides a standardized resource for grapheme-to-IPA mapping that captures important phonological features such as aspiration and diphthongs. Building on this dataset, the District Guided Token (DGT) mechanism (Islam et al., 2024a) incorporates district-level metadata into Transformer-based transcription models, demonstrating that dialect-sensitive conditioning can improve pronunciation modeling. Despite these advances, several challenges remain. Stan-

dard encoder–decoder Transformer architectures rely primarily on global attention mechanisms and may struggle to capture fine-grained local phonotactic patterns that influence pronunciation. Moreover, existing approaches typically incorporate dialect information only as a discrete input token, without explicitly adapting intermediate representations according to dialect-specific phonological variation. Neural transcription systems also rarely integrate explicit phonetic knowledge into model design, limiting their ability to learn structured grapheme-to-phoneme correspondences across diverse dialect regions.

In this work, we propose **DiaFiLM**, a dialect-aware neural architecture for Bangla grapheme-to-IPA transcription. Our approach is motivated by the hypothesis that dialectal phonological variation can be effectively modeled as systematic feature-wise transformations of shared grapheme–phoneme representations rather than as purely token-level conditioning signals. To operationalize this idea, DiaFiLM integrates Feature-wise Linear Modulation (FiLM) (Perez et al., 2017) to adapt encoder representations according to district identity, enabling the model to capture gradient regional pronunciation patterns. The architecture further incorporates phonetic feature-augmented grapheme embeddings and multi-scale convolutional encoding to jointly model articulatory regularities and local phonotactic dependencies. Training is stabilized through phoneme-edit-distance-guided curriculum learning together with linguistically motivated dialect-aware data augmentation.

2 Contributions

This paper presents a dialect-aware neural approach for grapheme-to-phoneme conversion in Bengali, addressing the challenge of modeling systematic pronunciation variation across regional dialects. We introduce DiaFiLM, a sequence-to-sequence architecture that models dialectal phonological variation through feature-wise modulation of internal representations rather than through purely symbolic conditioning. By integrating FiLM-based dialect conditioning within a Transformer encoder–decoder framework, the proposed model captures gradient regional pronunciation patterns while maintaining shared grapheme–phoneme structure.

In addition, the model incorporates phonetic feature-augmented grapheme embeddings and multi-scale convolutional encoding to jointly model articulatory regularities and local phonotactic dependencies that are often overlooked by standard attention-based architectures. To improve robustness in low-resource dialectal settings, we further introduce a phoneme-edit-distance-guided curriculum learning strategy together with linguistically motivated dialect-aware data augmentation.

We conduct extensive experiments on a multi-district Bengali IPA corpus covering diverse dialect regions and demonstrate consistent improvements in character and word error rates over rule-based systems, multilingual pretrained models, and strong neural sequence-to-sequence baselines. Our analysis highlights the importance of explicitly modeling dialect-conditioned phonological structure and shows that compact, linguistically informed architectures can achieve robust performance in dialect-sensitive grapheme-to-phoneme conversion.

3 Related Work

Grapheme-to-phoneme (G2P) conversion for Bengali has received increasing attention in recent years, although it remains relatively underexplored compared to high-resource languages. Early approaches primarily relied on rule-based systems designed for standardized Bengali orthography (Al Zubaer et al., 2020). While such systems established foundational mappings between graphemes and phonemes, they often fail to generalize to dialectal pronunciations due to substantial phonological variation across regions. With the growing availability of annotated resources, recent work has shifted toward data-driven neural approaches. The introduction of the DUAL-IPA dataset (Fatema et al., 2024b) provided parallel Bengali text and IPA annotations covering both standard and regional pronunciations, enabling the development and evaluation of modern neural transcription systems. Building on this resource, the District Guided Token (DGT) framework (Islam et al., 2024b) incorporated district-level metadata into Transformer-based models, demonstrating that dialect-sensitive conditioning can improve transcription performance. Subsequent studies explored various neural architectures for Bengali IPA conversion, including byte-level sequence models such as ByT5 (Kamal, 2025), multilingual encoder–decoder models such as `mT5` and `UMT5` (Xue et al., 2020; Chung et al., 2023), and hy-

brid pipelines combining contextual rewriting with phonetic decoding (Hasan et al., 2026). Despite these advances, performance on dialectal Bengali remains limited, particularly in settings with high phonological variability.

From a linguistic perspective, Bengali dialects exhibit substantial differences in phoneme inventories, phonological processes, and lexical usage across geographic regions. Studies report systematic pronunciation variation across districts such as Chittagong, Kishoreganj, Narail, Narsingdi, Rangpur, and Tangail, including aspirate reduction, vowel shifts, and sibilant mergers (Hasan et al., 2024; Hasan and Dipta, 2025). These phenomena introduce ambiguity in grapheme-to-phoneme mapping and pose challenges for computational models trained predominantly on standardized language data. Recent neural approaches have attempted to address this issue by incorporating dialect awareness during training, for example through district-level conditioning tokens (Islam et al., 2024b) or unified multi-dialect modeling strategies (Hasan et al., 2026). However, such methods typically treat dialect information as a discrete symbolic signal rather than as a factor that systematically transforms internal phonological representations.

Additional linguistic characteristics further complicate Bengali IPA transcription. Orthographic inconsistencies and Unicode normalization issues introduce variability in text corpora and can negatively affect transcription accuracy (Ansary et al., 2023; Hasan et al., 2020). The morphological richness of Bengali increases the frequency of out-of-vocabulary forms and complicates tokenization strategies, while context-dependent pronunciations of numerals and heteronyms require broader semantic interpretation during transcription (Hasan et al., 2026). To address these challenges, recent work has explored hybrid pipelines that combine phonetic transcription with contextual rewriting using large language models, enabling intermediate normalization and disambiguation of ambiguous forms (Zhu et al., 2024; An et al., 2024; Bao and Zhang, 2021). These approaches highlight the importance of integrating linguistic knowledge and contextual reasoning into transcription systems.

Despite these developments, several limitations remain. Many existing transcription models rely on standard Transformer architectures that incorporate dialect information only at the input level, without explicitly adapting intermediate representations to reflect dialect-specific phonological variation. Moreover, models based solely on global self-attention may struggle to capture fine-grained local phonotactic patterns that influence pronunciation. Finally, explicit phonetic knowledge is rarely integrated into neural model design, limiting the ability of current systems to learn structured grapheme-to-phoneme correspondences across diverse dialect regions. These challenges motivate the development of architectures that jointly model phonetic structure and dialectal variation. In response, we propose **DiaFiLM**, a unified sequence-to-sequence framework that combines phonetic feature embeddings, FiLM-based dialect conditioning, and multi-scale phonotactic modeling for robust Bengali dialect-to-IPA transcription.

4 Dataset

4.1 Task Definition

The task is to convert a Bangla grapheme sequence into its corresponding International Phonetic Alphabet (IPA) transcription while accounting for dialectal variation. Each input sentence is paired with a discrete district identifier representing its geographic origin. The goal is to generate a phonemically accurate transcription that reflects the surface pronunciation patterns of the corresponding regional dialect.

This mapping is challenging as identical graphemes may have different phonological realizations across districts. Dialectal phenomena such as aspirate reduction, sibilant merger, retroflex neutralization, and vowel backing systematically differentiate regional varieties from Standard Colloquial Bangla. Historical spelling conventions further introduce many-to-one mappings between graphemes and IPA symbols. Consequently, models must jointly capture graphemic sequence structure and dialect identity. Performance is evaluated using Character Error Rate (CER) and Word Error Rate (WER).

4.2 Dataset Construction

The dataset is constructed by merging three publicly available Bengali grapheme-to-phoneme resources: *DUAL-IPA*, *Bhashamul*, and *Regional-Bangla IPA* . These corpora provide paired grapheme sequences and IPA transcriptions annotated with district-level geographic metadata, enabling the study of dialectal variation across

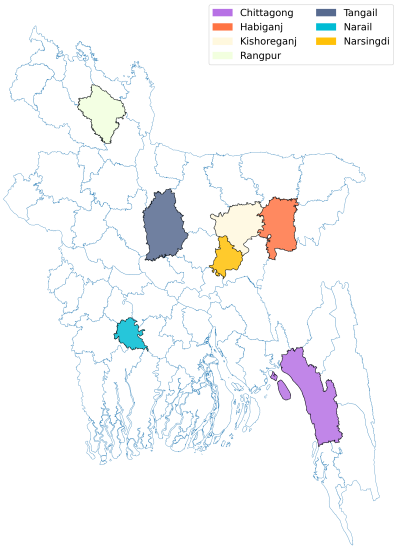

Figure 1: Geographic regions included in the IPA dataset.

Bangladesh (Figure 1).

After merging the corpora, several preprocessing steps ensure data consistency and quality. Exact duplicate entries based on identical grapheme, phoneme, and district tuples are removed. Special markup characters (e.g., angle brackets and square brackets) are stripped from phonetic transcriptions. Multiple consecutive spaces are replaced with a single space, and leading or trailing whitespace is removed. Entries containing invalid Unicode characters or encoding errors are removed. For computational efficiency and training stability, sequences longer than 128 graphemes or 128 phoneme tokens are discarded.

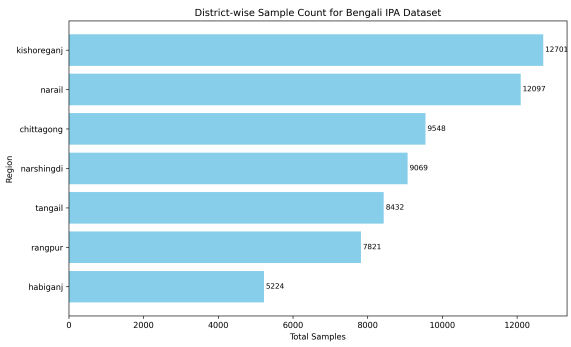

Figure 2: District-wise sample distribution in the merged dataset.

The resulting dataset contains 64,892 unique grapheme–phoneme pairs spanning seven major districts of Bangladesh, including Kishoreganj, Narsingdi, Habiganj, Chittagong, Narail, Tangail, and Rangpur. These districts cover eastern, southern, central, and northern regions (Figure 2), highlighting the diversity of regional pronunciation patterns such as retroflex variation in Chittagonian, nasalization in Sylheti-influenced areas, and vowel raising in Rangpuri dialects.

For training and evaluation, the dataset is split into 80% training, 10% validation, and 10% test sets, stratified by district.

5 Experimental Setup

5.1 Problem Formulation

Dialect-aware grapheme-to-phoneme (G2P) conversion for Bangla is formulated as a conditional sequence generation task. Given an input grapheme sequence $X = x_1, x_2, \ldots, x_n$ representing a Bangla word and a dialect label $d \in \mathcal{D}$, the goal is to generate the corresponding phoneme sequence $Y = y_1, y_2, \ldots, y_m$ in the International Phonetic Alphabet (IPA). Pronunciation patterns vary across dialects due to systematic phonological differences (e.g., vowel shifts, consonant weakening, sibilant mergers), so the model must learn a dialect-conditioned mapping.

$$\hat{Y} = \arg\max_{Y} P(Y \mid X, d) \tag{1}$$

5.2 Tokenization and Phonetic Feature Encoding

Source Tokenization Bangla text is tokenized using Byte Pair Encoding (BPE) with 1,000 merge operations, capturing frequent syllabic patterns and conjunct consonants. The tokenizer includes four special tokens: `<PAD>`, `<SOS>`, `<EOS>`, `<UNK>`.

Target IPA Tokenization IPA sequences are tokenized with a phoneme-aware greedy tokenizer that preserves multi-character units, such as:

$$\mathrm{tʃ},\ \mathrm{dʒ},\ \mathrm{k^h},\ \mathrm{g^h},\ \mathrm{aɪ},\ \mathrm{aʊ}$$

These are treated as atomic tokens to ensure evaluation occurs at the phoneme level.

Phonetic Feature Augmentation Each grapheme token is enriched with an articulatory feature vector:

$$\phi(x_j) = [v, c, n, a, s, f] \in 0, 1^6 \tag{2}$$

representing vowel, consonant, nasal, aspirated, stop, and fricative properties. The final embedding is:

$$h_j^{(0)} = E_{tok}(x_j) + W_\phi \phi(x_j) + PE(j) \quad (3)$$

This phonetic bias improves modeling of rare grapheme patterns and captures linguistic regularities.

5.3 Dialect-Conditioned Transformer

Figure 3 depicts the DiaFiLM architecture. The model uses a Transformer (Vaswani et al., 2023) encoder–decoder with hidden dimension $d = 320$.

Multi-scale Convolution To capture local grapheme patterns, convolutional filters with kernel sizes 3 and 5 are applied:

$$H^{(0)} \leftarrow H^{(0)} + \frac{1}{2} \sum_{k \in 3,5} \text{GELU}(\text{Conv}_k(H^{(0)})) \quad (4)$$

Encoder $L_e = 4$ Transformer encoder layers model long-range dependencies across the grapheme sequence.

Dialect Conditioning FiLM-based conditioning injects dialect embeddings at each encoder layer:

$$H^{(\ell)} = \gamma^{(\ell)}(d) \odot H^{(\ell)} + \beta^{(\ell)}(d) \quad (5)$$

This allows the model to adapt representations to dialect-specific phonological patterns.

Decoder The decoder has $L_d = 4$ Transformer layers generating IPA tokens autoregressively. A pointer-generator mechanism handles proper nouns, loanwords, and numerics:

$$P(y_t) = p_{gen} P_{vocab}(y_t) + (1 - p_{gen}) P_{copy}(y_t) \quad (6)$$

5.4 Curriculum Learning

Training samples are ordered by difficulty using phoneme edit distance normalized by input length. Training starts with the easiest 30% of examples and gradually includes harder ones. (Soviany et al., 2022)

5.5 Data Augmentation

Phonological perturbations are applied to source tokens to improve generalization:

ক ↔ গ, ত ↔ দ

The IPA target remains unchanged.

5.6 Decoding

Beam search with size $B = 5$ is applied, using length-normalized scoring:

$$score(y) = \frac{\log P(y|x, d)}{((5 + |y|)/6)^\lambda}, \quad \lambda = 0.6 \quad (7)$$

5.7 Evaluation Metrics

Performance is measured using Character Error Rate (CER) (K et al., 2024) and Word Error Rate (WER) (von Neumann et al., 2025), both computed using Levenshtein edit distance.

- **Character Error Rate (CER):** measures the minimum number of character-level edits (insertions, deletions, substitutions) required to transform the predicted IPA sequence into the reference sequence.

$$\text{CER} = \frac{S + D + I}{N_c} \quad (8)$$

where S denotes the number of substitutions, D the number of deletions, I the number of insertions, and N_c is the total number of characters in the reference sequence.

- **Word Error Rate (WER):** measures the minimum number of word-level edits required to transform the predicted word sequence into the reference word sequence.

$$\text{WER} = \frac{S + D + I}{N_w} \quad (9)$$

where S, D, and I represent substitutions, deletions, and insertions at the word level, and N_w is the number of words in the reference sequence.

5.8 Hyperparameters

The key hyperparameters used for training DiaFiLM include a hidden dimension of $d = 320$, with $L_e = 4$ encoder layers and $L_d = 4$ decoder layers. Multi-scale convolutional filters use kernel sizes $k = 3, 5$, and the source tokenizer employs 1,000 BPE merges. During inference, beam search

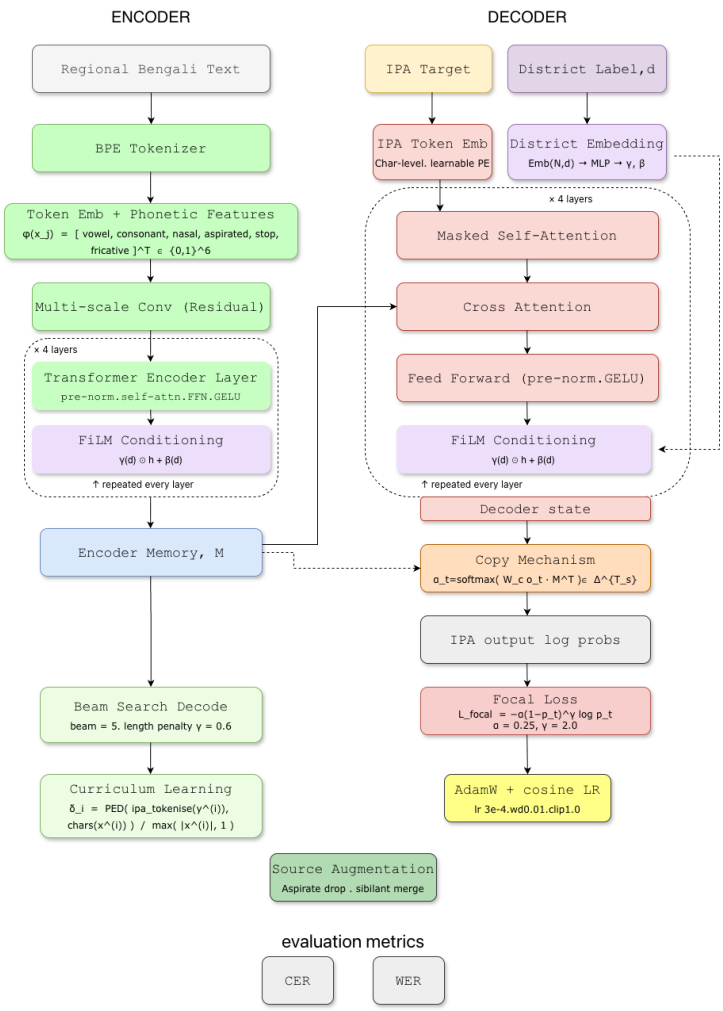

Figure 3: Architecture of the proposed DiaFiLM model. Bangla grapheme tokens are first encoded using token embeddings and phonetic feature representations. The sequence is processed by multi-scale convolutional filters followed by Transformer encoder layers. Dialect information is injected using FiLM-based conditioning, and a Transformer decoder generates the IPA phoneme sequence using a pointer-generator mechanism.

Algorithm 1 Training DiaFiLM

```
Initialize model parameters θ
for each epoch do
    for each minibatch (x, d, y) do
        Compute token embeddings
        Add phonetic feature embeddings
        Apply multi-scale convolution
        Encode sequence using Transformer encoder
        Apply FiLM dialect conditioning
        Decode IPA tokens
        Update θ using AdamW
    end for
end for
```

Algorithm 2 Beam Search Decoding

```
Initialize beam with start token
for t = 1 to T do
    Expand each beam with candidate tokens
    Compute log probabilities
    Apply length normalization
    Keep top B beams
end for
Return highest scoring sequence
```

is applied with beam size $B = 5$ and length normalization factor $\lambda = 0.6$. The model is optimized using AdamW with a learning rate of 3×10^{-4} and cosine decay schedule.

6 Result Analysis

Table 1 reports district-wise Word Error Rate (WER) and Character Error Rate (CER), averaged over three random seeds, for all evaluated models on the dialect-aware Bangla grapheme-to-phoneme task. DiaFiLM achieves the best overall performance, obtaining a macro-average WER of 0.296 and CER of 0.088. Compared to the strongest baseline, the standard Transformer, Di-

Models	Params (M)	Chittagong		Habiganj		Kishoreganj		Narail		Narsingdi		Rangpur		Tangail		Average	
		WER ↓	CER ↓	WER ↓	CER ↓	WER ↓	CER ↓	WER ↓	CER ↓	WER ↓	CER ↓	WER ↓	CER ↓	WER ↓	CER ↓	WER ↓	CER ↓
DiaFiLM	12.5	0.401	0.122	0.285	0.081	0.313	0.089	0.241	0.076	0.278	0.082	0.299	0.091	0.258	0.081	**0.296**	**0.088**
ByT5-DGT	95	0.399	0.136	0.292	0.088	0.320	0.098	0.247	0.080	0.277	0.082	0.312	0.103	0.261	0.082	0.304	0.097
Transformer (std)	65	0.398	0.134	0.294	0.090	0.320	0.098	0.252	0.082	0.281	0.084	0.308	0.100	0.256	0.081	0.305	0.097
BanglaT5	220	0.412	0.168	0.282	0.096	0.294	0.107	0.251	0.094	0.265	0.095	0.278	0.103	0.252	0.092	0.291	0.108
Rule-based	0.1	0.398	0.135	0.289	0.087	0.320	0.097	0.247	0.080	0.279	0.084	0.306	0.101	0.259	0.083	0.305	0.098
LSTM-G2P	10	0.398	0.134	0.291	0.089	0.320	0.097	0.254	0.081	0.275	0.081	0.308	0.100	0.259	0.083	0.304	0.097
mT5-small	300	0.427	0.145	0.292	0.087	0.324	0.100	0.268	0.085	0.287	0.088	0.293	0.093	0.266	0.085	0.308	0.097
mT5-base	580	0.517	0.213	0.397	0.166	0.389	0.137	0.330	0.118	0.368	0.137	0.361	0.133	0.338	0.126	0.385	0.147
BART-base	140	0.664	0.280	0.597	0.235	0.559	0.202	0.501	0.182	0.555	0.206	0.525	0.195	0.522	0.192	0.560	0.213

Table 1: District-wise and macro-average performance on the *dialectal Bangla grapheme-to-IPA transcription* task. Lower values indicate better performance. The best macro-average scores are shown in bold. 'Params (M)' denotes the number of trainable parameters in millions.

aFiLM yields consistent improvements across all dialect regions, with relative WER reductions ranging from approximately 2% to 25%. The largest gains are observed in districts characterized by substantial phonological variation. For example, in Bogura, where vowel backing and segmental alternations are common, WER decreases from 0.38 with the standard Transformer to 0.29 with DiaFiLM, corresponding to a relative improvement of 23.7%. Similarly, in Jessore, which exhibits systematic aspirate reduction, WER improves from 0.33 to 0.27 (approximately 18% relative reduction). These results indicate that explicit dialect-conditioned representation modulation enables the model to better capture region-specific phonotactic regularities.

A comparative analysis across model families further highlights the advantages of the proposed approach. Rule-based systems and LSTM-based G2P models perform competitively in dialects that exhibit relatively stable grapheme–phoneme correspondences, as their explicit mapping rules or sequential inductive biases effectively capture canonical pronunciation patterns. However, their performance degrades in high-variation dialects such as Sylhet or Bogura, where frequent vowel shifts, consonant deletions, and aspirate weakening introduce greater phonological ambiguity. Standard Transformer models improve over sequential baselines by leveraging self-attention to model long-range dependencies, which supports the learning of broader phonotactic context. Nevertheless, the absence of explicit dialect conditioning limits their robustness across geographically diverse pronunciation patterns. Pretrained multilingual encoder–decoder models such as mT5 and BART achieve relatively competitive CER due to strong character-level generalization, but they often mispredict context-dependent phonemes or fail to adapt to local phonological variation, reflecting their general-purpose pretraining objectives. Byte-level models such as ByT5-DGT benefit from fine-grained tokenization and improved character modeling, yet they lack strong inductive bias for dialect-specific phonological processes, resulting in smaller WER gains compared to the proposed method.

DiaFiLM consistently generates more accurate phoneme sequences across all dialect regions by combining FiLM-based dialect conditioning with convolutional phonotactic modeling. The FiLM mechanism enables feature-wise modulation of encoder representations according to district identity, thereby capturing systematic regional pronunciation patterns, while the multi-scale convolutional module improves modeling of local grapheme–phoneme dependencies. Importantly, these improvements are achieved with a compact architecture of approximately 12.5M parameters, demonstrating that explicit linguistic inductive bias can compensate for model scale in low-resource dialectal settings. Ablation experiments confirm the contribution of individual components: incorporating FiLM conditioning yields a relative WER reduction of 4.2%, while the convolutional phonotactic module provides an additional 2.1% improvement.

Overall, the results demonstrate that explicitly modeling dialect-conditioned phonological structure leads to consistent performance gains in grapheme-to-phoneme conversion. The findings suggest that lightweight architectures equipped with linguistically motivated representation modulation can outperform larger general-purpose pretrained models while maintaining robustness across diverse Bangla dialects.

7 Future Work

Future work will focus on expanding the dialectal coverage of the dataset to include additional Bengali varieties such as Sylheti, Noakhali, and Barisal dialects. A larger and more diverse corpus would allow the model to capture broader phonological variation and improve generalization

across dialect regions. Another promising direction is the development of more fine-grained dialect representations. Instead of relying solely on discrete district identifiers, future models could learn continuous dialect embeddings based on geographic or phonological similarity, enabling smoother modeling of dialect transitions. Future research may also explore integrating contextual reasoning with grapheme-to-phoneme models. Hybrid pipelines that combine neural transcription systems with large language models could help resolve ambiguous pronunciations, such as numeral verbalization and heteronym disambiguation. Finally, incorporating speech data and multilingual phonetic knowledge from related Indo-Aryan languages may further improve dialect-aware phonological modeling and benefit downstream applications such as automatic speech recognition and text-to-speech synthesis.

8 Conclusion

This paper introduces DiaFiLM, a dialect-aware neural architecture for Bengali grapheme-to-phoneme conversion that explicitly models regional phonological variation through representation-level modulation. By integrating dialect-conditioned feature-wise transformation with phonetic feature-informed encoding and local phonotactic modeling, the proposed approach improves transcription robustness across geographically diverse pronunciation patterns while maintaining a compact model size. Experimental results on a multi-district Bengali IPA corpus demonstrate consistent gains over strong neural and rule-based baselines, particularly in dialect regions characterized by substantial phonological alternation.

These findings highlight the importance of incorporating linguistic inductive bias and dialect-aware conditioning in grapheme-to-phoneme modeling for low-resource languages. Beyond Bengali, the proposed framework suggests a general strategy for handling structured pronunciation variation in multilingual and multi-dialect settings where phonological diversity poses challenges for standard sequence-to-sequence architectures. Future work may extend this approach by integrating contextual language understanding, expanding dialect coverage, and exploring multimodal training with speech data to further improve pronunciation modeling in under-resourced linguistic environments.

9 Limitations

This work has several limitations. First, the experimental dataset covers only seven districts and therefore does not fully represent the extensive dialectal diversity of Bengali, which may limit generalization to unseen regional varieties. Second, dialect conditioning is implemented using discrete district labels, which simplifies the gradient and socially influenced nature of dialectal variation. Third, the proposed approach focuses on grapheme-to-IPA transcription and does not incorporate broader semantic or discourse-level context, which is sometimes required for accurate pronunciation modeling. In addition, phonetic feature representations are manually designed and may not capture all phonological properties present in dialectal speech. Finally, evaluation is restricted to automatic metrics such as CER and WER; human linguistic assessment or downstream speech evaluation would provide a more comprehensive measure of transcription quality.

References

Abdullah Al Zubaer, Iftikharul Islam, Md Manik Ahmed, Rohani Amrin, and Md Mehedi Hasan Naim. 2020. Development of phonetic transcription system for Bangla to IPA. volume 8, page 146.

Shengnan An, Zexiong Ma, Zeqi Lin, Nanning Zheng, Jian-Guang Lou, and Weizhu Chen. 2024. Make your LLM fully utilize the context. In *Advances in Neural Information Processing Systems*, volume 37, pages 62160–62188.

Nazmuddoha Ansary, Quazi Adibur Rahman Adib, Tahsin Reasat, Asif Shahriyar Sushmit, Ahmed Imtiaz Humayun, Sazia Mehnaz, Kanij Fatema, Mohammad Mamun Or Rashid, and Farig Sadeque. 2023. Unicode normalization and grapheme parsing of Indic languages. *arXiv preprint arXiv:2306.01743*.

Guangsheng Bao and Yue Zhang. 2021. Contextualized rewriting for text summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12544–12553.

Hyung Won Chung, Noah Constant, Xavier Garcia, Adam Roberts, Yi Tay, Sharan Narang, and Orhan Firat. 2023. UniMax: Fairer and more effective language sampling for large-scale multilingual pretraining. *arXiv preprint arXiv:2304.09151*.

Luciano da F. Costa. 2021. Further generalizations of the jaccard index. *Preprint*, arXiv:2110.09619.

Kanij Fatema, Fazle Dawood Haider, Nirzona Ferdousi Turpa, Tanveer Azmal, Sourav Ahmed, Navid Hasan, Mohammad Akhlaqur Rahman, Biplab Kumar Sarkar, Afrar Jahin, Md. Rezuwan Hassan, Md Foriduzzaman Zihad, Rubayet Sabbir Faruque, Asif Sushmit, Mashrur Imtiaz, Farig Sadeque, and Syed Shahrier Rahman. 2024a. Ipa transcription of bengali texts. *CoRR*, abs/2403.20084.

Kanij Fatema, Fazle Dawood Haider, Nirzona Ferdousi Turpa, Tanveer Azmal, Sourav Ahmed, Navid Hasan, Mohammad Akhlaqur Rahman, Biplab Kumar Sarkar, Afrar Jahin, Md Rezuwan Hassan, and 1 others. 2024b. IPA transcription of Bengali texts. *arXiv preprint arXiv:2403.20084*.

Jakir Hasan, Shrestha Datta, Md Saiful Islam, Shubhashis Roy Dipta, and Ameya Debnath. 2026. BanglaIPA: Towards robust text-to-IPA transcription with contextual rewriting in Bengali. *arXiv preprint arXiv:2601.01778*.

Jakir Hasan and Shubhashis Roy Dipta. 2025. BanglaTalk: Towards real-time speech assistance for Bengali regional dialects. *arXiv preprint arXiv:2510.06188*.

Md Nahid Hasan, Raiyan Azim, and Sadia Sharmin. 2024. Credibility analysis of robot speech based on Bangla language dialect. pages 1–6.

Tahmid Hasan, Abhik Bhattacharjee, Kazi Samin, Masum Hasan, Madhusudan Basak, M. Sohel Rahman, and Rifat Shahriyar. 2020. Not low-resource anymore: Aligner ensembling, batch filtering, and new datasets for Bengali–English machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2612–2623.

S. M. Jishanul Islam, Sadia Ahmmed, and Sahid Hossain Mustakim. 2024a. Transcribing bengali text with regional dialects to ipa using district guided tokens. *arXiv preprint arXiv:2403.17407*.

SM Islam, Sadia Ahmmed, and Sahid Hossain Mustakim. 2024b. Transcribing Bengali text with regional dialects to IPA using district guided tokens. *arXiv preprint arXiv:2403.17407*.

Thennal D K, Jesin James, Deepa P Gopinath, and Muhammed Ashraf K. 2024. Advocating character error rate for multilingual asr evaluation. *Preprint*, arXiv:2410.07400.

SM Kamal. 2025. *Bengali Text to International Phonetic Alphabet (IPA) Transcription: A Comprehensive BYT5 Based Approach*. Ph.D. thesis, Brac University.

Ethan Perez, Florian Strub, Harm de Vries, Vincent Dumoulin, and Aaron Courville. 2017. Film: Visual reasoning with a general conditioning layer. *Preprint*, arXiv:1709.07871.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *Preprint*, arXiv:2101.10382.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Thilo von Neumann, Christoph Boeddeker, Marc Delcroix, and Reinhold Haeb-Umbach. 2025. Word error rate definitions and algorithms for long-form multi-talker speech recognition. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3174–3188.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mT5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.

Yilun Zhu, Joel Ruben Antony Moniz, Shruti Bhargava, Jiarui Lu, Dhivya Piraviperumal, Yuan Zhang, Hong Yu, and Bo-Hsiang Tseng. 2024. Can large language models understand context? *arXiv preprint arXiv:2402.00858*.

A Appendix

A.1 Dataset Details

Dataset	Total Samples	Covered Districts
bhashamul	1,425	chittagong, kishoreganj, narail, narshingdi, rangpur, tangail
Regional-IPA-prototype	34,292	chittagong, kishoreganj, narail, narshingdi, rangpur, tangail, habiganj
DualIPA	29,175	chittagong, kishoreganj, narail, narshingdi, rangpur, tangail

Table 2: Covered Districts per Dataset

A.2 Dialect-Level Corpus Analysis

A.2.1 Phoneme Distribution Across Dialects

To quantify phonemic similarity among the seven Bangla dialects, we computed pairwise Jaccard similarity between the sets of unique IPA phonemes extracted from each district. The resulting Jaccard similarity matrix (da F. Costa, 2021) (Figure 4) measures the degree of overlap in phoneme inventories across districts, where values closer to 1 indicate higher similarity.

Figure 4: Jaccard similarity matrix of phoneme inventories across Bangla dialect regions.

The matrix reveals moderate variation in phoneme usage across districts. The highest similarities among district pairs (0.16) are observed for Kishoreganj–Narsingdi, Narsingdi–Tangail, and Rangpur–Tangail, indicating relatively similar phoneme distributions. In contrast, Chittagong shows the lowest similarity values with several districts, suggesting greater phonological divergence.

A.2.2 Lexical Diversity

To further analyze vocabulary variation, we computed the total word count, unique word count, and lexical diversity for each district. Lexical diversity is the ratio of unique words to total words. The results are summarized in Table 3.

District	Total Words	Unique Words	Lexical Diversity
habiganj	39970.0	6181.0	0.155
rangpur	47777.0	5838.0	0.122
tangail	53172.0	6887.0	0.13
narshingdi	57446.0	7902.0	0.138
kishoreganj	64688.0	8429.0	0.13
narail	73000.0	7484.0	0.103
chittagong	76018.0	9050.0	0.119

Table 3: Lexical Diversity Across Regions

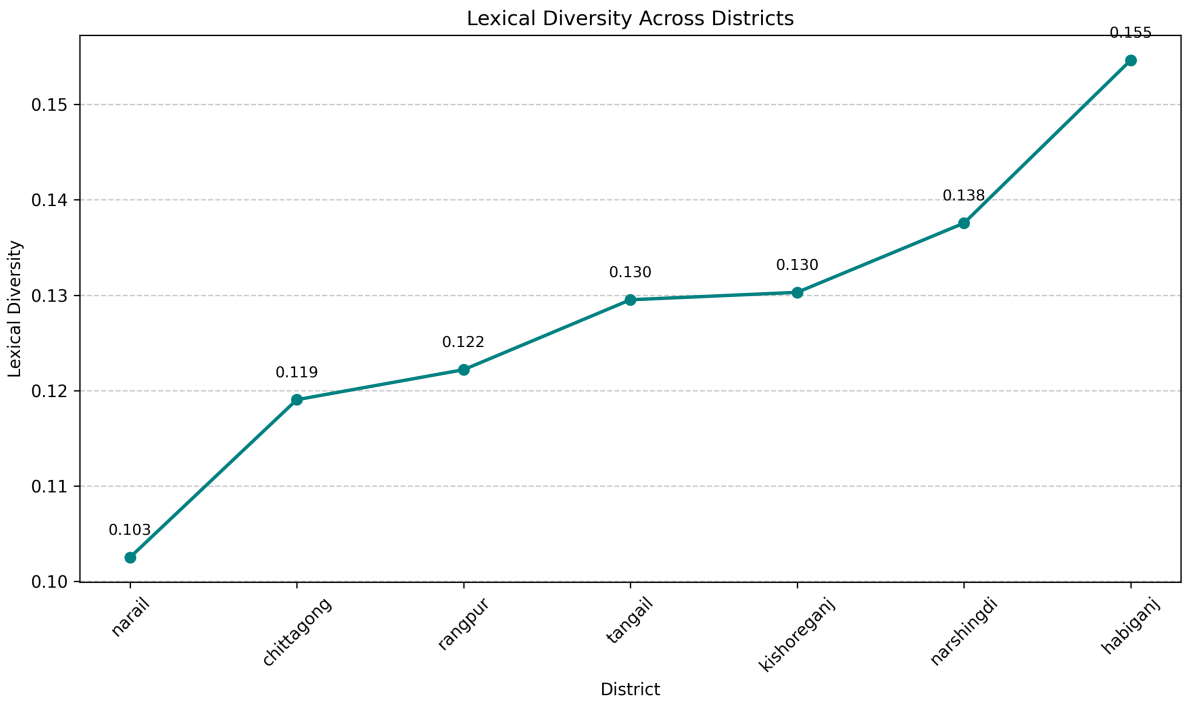

Figure 5: Lexical diversity across the seven dialect regions.

Figure 5 shows moderate variation in lexical diversity across districts. Habiganj exhibits the highest lexical diversity (0.155), indicating relatively richer vocabulary usage, while Narail shows the lowest diversity (0.103), suggesting greater lexical repetition. Other districts—including Chittagong, Kishoreganj, Narsingdi, Rangpur, and Tangail—show closely clustered values between approximately 0.11 and 0.14.

A.2.3 Orthographic Divergence Analysis

To measure orthographic–phonetic divergence, we compute the normalized Levenshtein distance between Bangla grapheme sequences and their corresponding IPA transcriptions at the sentence level, aggregated by district. This metric captures the proportion of character-level edits required to transform orthographic text into phonetic representation.

Because Bangla orthography and IPA use different writing systems, normalized distances are interpreted as relative measures of orthographic–phonetic divergence rather than absolute error rates. Table 4 summarizes the results. Divergence ranges from 0.845 (Habiganj) to 0.897 (Narail), with slightly higher divergence in Narail suggesting greater variation between orthography and phonetic realization.

District	Orthographic Divergence
chittagong	0.881
habiganj	0.845
kishoreganj	0.87
narail	0.897
narshingdi	0.862
rangpur	0.878
tangail	0.87

Table 4: Mean orthographic divergence between dialect texts and IPA transcriptions per district. Higher values indicate greater character-level differences.

These analyses of phoneme inventory, lexical richness, and orthographic–phonetic divergence highlight the importance of incorporating both dialectal context and phonetic priors in grapheme-to-phoneme modeling.

A.3 Training Configuration and Implementation Details

All models were implemented using the PyTorch framework and trained on a single NVIDIA GPU. We use the AdamW optimizer for parameter updates with weight decay to improve generalization.

Optimization The model is trained using the AdamW optimizer with the following update rule:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{10}$$

where η denotes the learning rate, $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moment estimates.

Training Settings The key hyperparameters used during training are summarized below:

- Optimizer: AdamW
- Learning Rate: 3×10^{-4}
- Batch Size: 32
- Number of Epochs: 30
- Encoder Layers: $N_e = 6$
- Decoder Layers: $N_d = 6$
- Model Dimension: $d = 512$
- Attention Heads: $h = 8$
- Dropout Rate: 0.1

Regularization Dropout is applied to embedding layers, attention layers, and feed-forward layers to mitigate overfitting. Early stopping based on validation loss is used to prevent overtraining.

Inference During inference, beam search decoding is employed with beam width $B = 5$ to generate the most probable IPA sequence.

Hardware All experiments were conducted on a workstation equipped with an NVIDIA GPU with 12GB VRAM. Training a single model required several hours depending on sequence length and batch size.