TOWARDS USING HETEROGENEOUS RELATION GRAPHS FOR END-TO-END TTS

Amrith Setlur[*,§], *Aman Madaan*[*,†], *Tanmay Parekh*[*,†], *Yiming Yang*[†], *Alan W Black*[†]

[†]Language Technologies Institute, [§]Machine Learning Department
Carnegie Mellon University

ABSTRACT

Neural models for end-to-end text-to-speech (TTS) synthesis are increasingly outperforming traditional approaches in statistical parametric speech synthesis. Speech generation in these neural models predominantly relies on using free-form text as the input modality. However, the earlier statistical parametric models were built on encoded phonetic and syntactic features. In this work, we explore the possibility of explicitly feeding deterministic linguistic structure to a neural TTS system in the form of Heterogeneous Relational Graphs (HRGs), an expressive formalism capable of representing phonetic and syntactic information. Specifically, we use Graph Convolutional Networks to learn structurally informed continuous representations of the HRGs, which can be seamlessly passed to the encoders of popular neural TTS models like TransformerTTS or Tacotron. Furthermore, our simple HRG based text-to-speech synthesis leverages the syntactic bias in HRGs as demonstrated by improvements in automated metrics and human evaluation on **i)** the single speaker dataset LJSpeech; **ii)** the multi-speaker dataset Arctic; and **iii)** out-of-domain test sets from the Blizzard challenge.[1]

Index Terms— text-to-speech, end-to-end neural TTS, Graph Convolutional Networks, Heterogeneous Relation Graphs.

1. INTRODUCTION

In end-to-end text-to-speech (TTS) synthesis, the objective is to learn a function that maps a given input text sequence to human-like audio (waveform). Early work on speech synthesis followed unit-selection based models [1] or multi-phase statistical parametric approaches [2] where constructing individual phases of feature extraction and waveform generation required extensive domain expertise. Recently, end-to-end neural models [3, 4, 5] for TTS have been shown to generate close to human-level audio when trained on large datasets ($\sim$ 24 hrs) of high-quality audio samples for long hours.

Neural TTS systems typically take free-form text as input and treat it as a sequence of characters. This form of input misses important linguistic information, which can be indicative of phonetic information, stress patterns, syllable structure, and word structure [6]. On the contrary, in the past, the benefits of incorporating linguistic structures have been made evident by statistical parametric synthesis models that explicitly encode the structure in terms of local neighborhoods around each phonetic/syntactic unit [2], for e.g., the number of phonemes in a word. Motivated by this, there has been an increasing interest in learning neural speech models [7, 8, 9], and representations [10] that can recover linguistically informed latent spaces capturing continuous sub-unit representations for syllables and phones. There has been extensive work in such structural encoding and even inferring latent phonetic relationships in a neural model [11, 12]. However, the utility of a generic representation framework that can encapsulate all forms of structured phonetic, acoustic, and syntactic features has not yet been investigated.

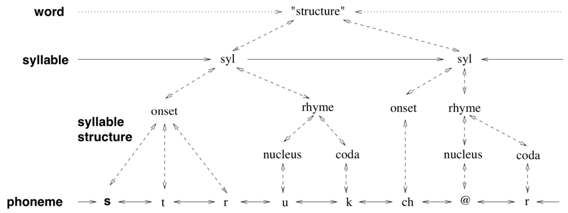

Fig. 1. Example of a Heterogeneous Relation Graph (HRG) that linguistically represents the pronunciation of the word "structure" [13].

In this work, we explore the possibility of feeding linguistic structure to neural TTS systems by revisiting the formalism of Heterogeneous Relation Graphs (HRGs) [13] as a generic expressive data structure that can represent complex phonetic and syntactic patterns via a graph. Heterogeneous Relation Graphs were designed as a data structure to enable statistical methods to jointly handle linguistically informed heterogeneous features including syntactic analysis, morphology, phonology, phonetics, prosody, articulatory control, and acoustics. Formally, an HRG is a graph $\mathcal{G}(V, E)$ where the set of vertices V includes syntactic, phonetic and acoustic units, and the set of edges E represents the various sequential (across time) and top-down (word $\rightarrow$ syllable $\rightarrow$ phoneme) dependencies that are present in any text utterance (Figure 1). Concretely, we pass the text input through the commonly used

[*] Equal contribution. Correspondences to asetlur@cs.cmu.edu.

[1] The code, trained models, and our dataset of HRGs will be released at https://github.com/ars22/GraphNeuralTTS/.

978-1-6654-3739-4/21/$31.00 ©2021 IEEE

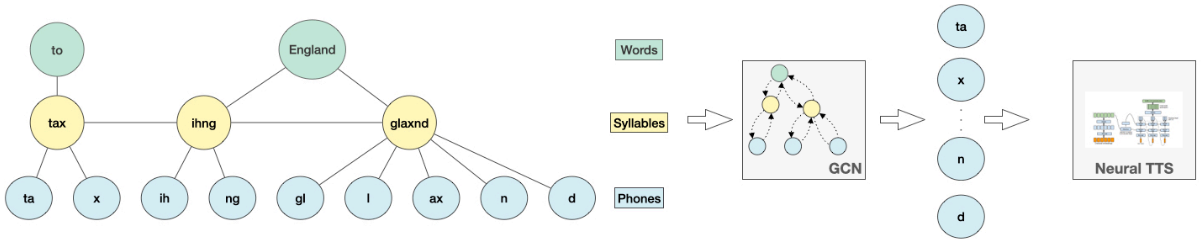

Fig. 2. The proposed pipeline for using Heterogeneous Relation Graphs (HRGs) to encode linguistic structure in the input text "to England". Graph Convolutional Networks (GCNs) are used to learn representations for words, syllables and phonemes in the HRG. The phoneme embeddings are extracted and passed to a neural TTS system like Tacotron or TransformerTTS which then output the corresponding mel spectrogram.

speech processing and synthesis tool: Festvox [6, 14], and obtain the corresponding HRG representation in a deterministic manner. After we get one such HRG for every text utterance in our dataset, our goal is to train a standard neural TTS model on this input modality of HRGs. For this work, we experiment with two widely popular neural architectures: **i)** Tacotron [3]; and **ii)** TransformerTTS [5], by suitably adapting them to accept as input, representations of HRGs, as opposed to the originally proposed character embeddings. To obtain a dense representation of the graph-structured linguistic information present in HRGs, we rely on recent advances in Graph Convolutional Networks (GCNs, [15]). A GCN refines the representation of each node by pooling features from its neighbors using a stack of convolutional layers. The GCN representations for each phoneme node in the HRG are then extracted and passed directly to the end-to-end neural model Tacotron/TransformerTTS. We exhibit the benefits of our approach through automated metrics and human evaluations, which indicate that neural models produce a higher quality of generated speech when fed with supervised linguistic information in the form of HRGs. In this work, we **do not claim** to propose a generic improvement (over all existing neural TTS models) in the output speech quality – but merely (re-)evaluate the advantage of a structural bias via the formalism of HRGs and further provide an easy way of mapping HRGs to neural representations that can be directly plugged into existing end-to-end models. Finally, we provide results in two additional scenarios: **i)** multi-speaker low data settings; and **ii)** out-of-domain speech synthesis where the model is trained on audio samples from non-fiction books and tested on conversations, and Semantically Unpredictable Sentences (SUS) from the Blizzard challenge [16].

The main contributions of our work are as follows: **i)** We revisit the utility of Heterogeneous Relation Graphs as a generic framework to represent phonetic/syntactic information in the input text utterances for state-of-the-art neural TTS models using Graph Convolution Networks ; **ii)** We empirically demonstrate the gains furnished by the linguistic bias in HRGs through experiments on the popular single-speaker dataset LJSpeech; **iii)** Through additional results in the more challenging multi-speaker low data setting (Arctic) as well as tests on out-of-domain text utterances from Blizzard, we further establish the usefulness of HRGs; **iv)** Finally, we provide some rationale for the observed improvements by analyzing the ability of HRGs to better predict duration (acoustic information) for each phoneme in the input.

2. RELATED WORKS

End-to-end neural models. One of the pioneering models in end-to-end synthesis is the deep neural network *Tacotron* [3] which encodes characters using an embedding lookup followed by a CBHG-based encoder and attention decoder to generate the output mel spectrograms. A vocoder like WaveNet [17] or the Griffin-Lim algorithm [18] is finally used to generate the waveform from the spectrogram output. This work was followed by rapid developments in end-to-end models [4, 19] and more recently, the multi-head attention based model TransformerTTS [5] has been shown to increase the training and inference efficiency, while improving the output speech quality by reducing errors caused by long term dependencies. We build on these advances by passing structured inputs (derived from HRGs) to these systems.

Structural inductive bias. Reinforcing the utility of phonetic and syntactic features in parametric speech synthesis, Ronanki et al. [20] proposed an approach that took the input linguistic features at their original timescales and preserved the relationships between words, syllables, and phones, improving the performance of their statistical system. The benefits of encoding prosodic words, intonational and prosodic phrase boundaries (PPH) for the task of Chinese speech synthesis was exhibited by Sun et al. [21]. Furthermore, Mametani et al. [22] conducted an extensive study on latent contextual features in end-to-end synthesis, comparing it with parametric TTS. Owing to the joint optimization of context and acoustic features, their work has shown that the encoder outputs reflect both linguistic and phonetic contexts, such as vowel reduction at phoneme level, lexical stress at the syllable level, and part-of-speech at the word level.

Graph based text-to-speech synthesis. Complementary to this paper, recent works [8, 9] have explored the benefits of structured/graph based inputs for neural TTS models. Sun et al. [8] explore the possibility of using word-character graphs to learn character representations that are fed to Tacotron. Liu et al. [9] map the input sentence to a syntax tree and use the syntactic relations between lexical tokens to derive syntactically motivated character embeddings for TTS attention mechanism. Although our results supplement their findings that structured inputs improve model generalizability, we introduce a more generic and rigorous structured representation framework of HRGs which can encode the relations between a wider set of syntactic, phonetic and (in some cases) acoustic units. HRGs can also be generated with different phone sets which can aid accent control, making our approach more widely applicable. Furthermore, since HRGs can be fetched using the widely recognized Festvox tool [6], they are easier to use and analyze.

3. METHODOLOGY

We begin with a brief overview of the expressivity of HRGs and describe the set of features we use to construct our inputs for the neural TTS models. We then provide details on extracting neural features from HRGs that can be directly fed to the encoders of the TTS models in their originally proposed form.

Heterogeneous Relation Graphs (HRGs). Taylor et al. [13] introduced HRGs as a self-contained schema to structurally represent the heterogeneous relationships present between syllables, words, and phonemes in a given text utterance. This formalism allows linguistic information to be encoded as graphs $\mathcal{G} = (V, E)$ s.t. V encompasses the sets of phonemes, syllables, words, and even phrase boundaries. The set of edges E allows for interesting relationships: for e.g., a *hierarchical tree* breaks words into phonemes (Figure 1). On the other hand, *multi-linear* lists allow associations between a sequence of intonational tones and corresponding syllables. Apart from features that are solely derived from the lexicon, HRGs can also have features obtained post Hidden Markov Model (HMM) alignment between phonemes and speech frames. Specifically, HRGs can encode timing information: the time (in ms) associated with each phoneme, along with $F0$ parameters and cepstra (stored as multi-linear lists). Finally, the heterogeneity of the relationships is exploited by having the same set of nodes V to be part of different views/graphs $\mathcal{G}_1 = (V, E_1), \mathcal{G}_2 = (V, E_2)$. For e.g., syllable nodes are part of a hierarchy between words and phonemes (E_1), while also being part of a metrical structure (E_2), which deciphers how much stress to lay on a subtree of syllables.

In this work, we exploit a very limited view of HRG's aforementioned capabilities. Specifically, we only extract linguistic information like phoneme and syllable sequences (Figure 2) that can be derived via a deterministic set of predefined rules in the Festvox [6] system. Since we don't use information drawn from the HMM alignment, we don't need the speech signal to obtain the linguistic structure for a given text utterance. We would like to highlight that restricting our approach to only use pre-alignment features benefits us in two ways: **i)** we allow for the usage of HRGs during inference (where we don't have the speech waveform); and **ii)** given a character sequence the corresponding HRG can be derived for it in a deterministic manner with negligible additional processing cost.

Processing HRGs for Neural TTS. While the node set V can contain a variety of information, we focus our experiments on using the following three types of nodes: i) words, ii) syllables, and iii) phonemes. Figure 2 shows a sample HRG for the sentence "to England". As shown, an HRG is a multi-granular representation of a sentence: a word node is linked to its syllables, and each syllable is further linked to the constituent phonemes. Apart from the top-down links, some nodes like syllables also have lateral links. The various levels in an HRG are characteristic of representations that capture both the surface level information in the form of words and phonemic information. Each of these result in different manifestations of acoustic units [20]. We posit that such a rich representation of the input text would help the model to learn from fewer hours of speech, and improve the quality of its generation by directly feeding off the rich linguistic structure in the input.

Graph Convolutional Networks. We use Graph Convolutional Networks (GCN, [15]) to learn rich node representations from the HRGs. GCNs can be used to learn node-level or graph-level representations (features) for tasks like node and graph classification. Our architecture consists of L layers of GCN. For a graph $\mathcal{G}(V, E)$, features $h_v^0 \in \mathbb{R}^k$ for each node $v \in V$ are randomly initialized from $\mathcal{N}(0, 0.3)$ (seed embedding). Each layer then refines node-features by aggregating information from its neighbors:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\frac{1}{|\mathbf{A}(v)|}\left(\mathbf{W}^l\mathbf{h}_v^l + \sum_{u\in\mathbf{A}(v)}\mathbf{W}^l\mathbf{h}_u^l\right)\right)$$

where σ is a non-linear activation function (*e.g.*, ReLU), $\mathbf{W}^l \in \mathbb{R}^{k\times k}$ is the GCN weight matrix for the l^{th} layer, and $\mathbf{A}(v)$ is the list of neighbors of a node v. Finally, the representation h_v^L for each phoneme node is fed as a sequence of input embeddings to the Tacotron/TransformerTTS model. This is in contrast with the original formulation, which encodes the input as a sequence of character embeddings.

Efficiency of HRGs. We note that apart from HRGs there are other ways of encoding structural features, for e.g, by using boundary markers (tokens) that indicate syllable or word boundaries in a sequence of phoneme tokens. However, such flattened representations of graphical relationships would be non-scalable (computationally heavy) since the length of the

Spectral analysis (Tacotron, TxTTS)	*pre-emphasis*: 0.97; *frame-length*: 50ms; *frame-shift*: 12.5ms; *window type*: Hann
GCN (Tacotron, TxTTS)	*num-layers* (L): 2; *hidden-size* (k): 256; *dropout-rate*: 0.3; *activation* (σ): ReLU
Encoder CBHG (Tacotron)	*Conv1D Bank*: K=16, conv-k-128-ReLU; *Max-pooling*: stride=1, width=2; *Conv1D Projections*: conv-3-128-ReLU → conv-3-128-Linear; *Highway net*: 4 layers of FC-128-ReLU; *Bidirectional GRU*: 128
Attention RNN (Tacotron)	*num-layers*: 1; *gru-hidden-size*: 256
Reduction Factor (Tacotron)	5 (Tacotron), 7 (Tacotron + Phoneme, Tacotron + HRG)

Table 1. Details on modified (from original implementation) or additional hyperparameters for Tacotron, TransformerTTS.

representation would grow exponentially with the depth of the input graph. In contrast, HRGs provide us with a more efficient way of encoding syntactic and phonetic features without flattening the hierarchical relations. Since graph convolutions are used to extract features for each node, the amortized processing time does not increase with additional relationships that may be represented by new edges in the HRG, and only scales linearly with the number of nodes in the graph (less if the graph convolutions are also parallelized).

4. EXPERIMENTAL SETUP

Dataset Details. For the in-domain setting, we train and test models on two datasets: **i)** *LJSpeech* [23] which contains 13,100 (11,790 train, 655 val and 655 test) high quality audio clips read from 7 different non-fictional books by a single female speaker with an American accent; and **ii)** *CMU Arctic*[2], a multi-speaker dataset comprising of 1132 sentences spoken by 18 different speakers (14,012 train, 785 val and 786 test) with three distinct accents – European, American, and Indian English. Furthermore, we use the *conversational (Conv)* and *Semantically Unpredictable Sentences (SUS)* test sets from the 2005 Blizzard challenge [16] to conduct an out-of-domain evaluation of models trained and validated on the LJSpeech corpus. The statistics for each dataset are provided in Table 2.

	Arctic	LJSpeech	Blizzard
Train	14,012	11,790	-
Validation	785	655	-
Test	786	655	50 (SUS) 50 (Conv)

Table 2. Data statistics for number of audio samples in the Arctic, LJSpeech and the Blizzard datasets.

Extracting HRGs. We use the Festvox toolkit [6] to extract an HRG for every given text utterance in a dataset. In the most general case, the Festvox steps include: **i)** initial text processing to build appropriate prompts for each utterance; **ii)** label alignment using Hidden Markov Models; **iii)** extraction of acoustic (duration), phonetic (syllables and phonemes) and syntactic (part-of-speech tags) features via ClusterGen [24]; and finally **iv)** construction of nodes (phonetic and syntactic units) and heterogeneous relations (in HRGs) from the features. Note that as mentioned in Section 3 we only use the pre-alignment features, to allow for the usage of HRGs at inference where we only have the text utterance (and not the corresponding speech file). We stress here that obtaining HRGs from a text-utterance is a lightweight and deterministic process (akin to POS-tagging). The official documentation[3] provides scripts to easily automate each of these steps with more details on the individual phases.

Model Details. Most of the hyperparameters for the base Tacotron and TransformerTTS models are borrowed as is from their respective original works [3] and [5]. In Table 1, we describe some of the modified and additional hyperparameter choices, including the hyperparameters of the GCN module. We use popular open-source implementations for Tacotron and TransformerTTS[4] and suitably modify them to use GCN representations of phoneme nodes withdrawn from HRGs. For both, we use the Adam [25] optimizer with an initial learning rate of $2e-3$ and update the learning rate based on validation mel loss. In all our experiments, we use the Griffin-Lim [18] algorithm (60 iterations) as the vocoder that maps mel spectrograms to waveforms. Note that in [5], TransformerTTS is trained for longer hours ($\sim 10^4$ epochs), but owing to computational constraints we train each model for a maximum of 500 epochs on each dataset. All the experiments use no more than two Nvidia RTX 2080 Ti GPU cards.

Baselines. We primarily experiment with three baseline models: **i)** *Tacotron:* The original Tacotron [3] model that is trained to output mel spectrograms for every input sequence of character embeddings, one for each character in the text utterance; **ii)** *Tacotron + Phoneme:* The Tacotron model which uses phoneme sequences instead of characters as the input; and **iii)** *TransformerTTS (TxTTS) + Phoneme* or *TxTTS*: Similar to Tacotron + Phoneme in terms of the input/output, but in principle based on the non-recurrent multi-head attention model [5]. Actually, we experimented with both phoneme and character embeddings for TxTTS, but did not find any difference in performance of the two variants for both Arctic and LJSpeech. Thus, we only use TxTTS with phoneme embeddings and any

[2] http://www.festvox.org/cmu_arctic/

[3] http://festvox.org/bsv

[4] Tacotron: https://github.com/keithito/tacotron, TransformerTTS: https://github.com/soobinseo/Transformer-TTS

reference to TxTTS means TxTTS + Phoneme. We refer to our proposed approaches as *Tacotron + HRG* and *TxTTS + HRG* where we use GCNs over heterogeneous relation graphs in conjunction with the baseline neural models of Tacotron and TxTTS respectively.

Human evaluation. We compute the mean-opinion-scores (MOS) over 9 annotators who are domain experts, and each annotator evaluates the quality (on a scale of 1-5) of 130 unseen audio samples for each method (40 each for Arctic and LJSpeech; 50 for Blizzard Conv) which allows us to get low variance performance estimates. The test set for Arctic comprises of 40 sampled utterances over 15 speakers (the test is unbiased since the number of test utterances for each speaker is proportional to their training samples). The MOS scores are averaged over all samples. The listening test samples for Arctic and LJSpeech are randomly selected from the unseen examples in the test splits for each, whereas the Blizzard test sets are obtained directly from the 2005 Blizzard Challenge [16]. Since the phoneme sequences in Blizzard SUS sentences are hard to predict – making the speech harder to recognize – the quality of the generated speech for these are judged on the word-error-rate (WER) between the ground truth text utterance and the annotators' transcription of the generated speech. Each of the 50 generated samples in the SUS set is transcribed by three expert human annotators and the final WER is averaged over these transcriptions.

Automated Evaluation. We use DTW-MCD[5] which uses dynamic time warping (DTW) to compute the minimum (over alignments) Mel Cepstral Distortion (MCD) between the generated and ground truth speech. Due to the lack of ground-truth speech, DTW-MCD is not computed for out-of-domain evaluations on the Blizzard sets.

5. RESULTS AND DISCUSSION

Main Results (in-domain). In Table 3 we note higher MOS scores and lower DTW-MCD values for our proposed approaches: Tacotron + HRG and TxTTS + HRG over the baseline models: Tacotron, Tacotron + Phoneme, and TxTTS on the LJSpeech and Arctic datasets[6]. The improvements are more pronounced for the TxTTS + HRG model, which improves over the TxTTS model by 16% for LJSpeech, and 54% for the Arctic dataset. Tacotron + HRG model improves over the Tacotron + Phoneme baseline with a significant average of 4.5% and 6.9% over LJSpeech and Arctic, respectively. We hypothesize that structural bias is more helpful in low-data settings since the model needs to learn generalizable features from fewer samples per speaker. This is further substantiated by our findings where higher gains are observed on the Arctic dataset (roughly 1 hour speech for each speaker) than the LJSpeech corpus ($\sim$ 24 hrs of high-quality audio from a single speaker). Note that in Table 3 the only case where the MOS scores have slightly overlapping confidence intervals are for Tacotron + Phoneme and Tacotron + HRG, but the difference in performance is still statistically significant since it has a very low p-value $\approx 1e-6$ for both Arctic and LJSpeech.

Model	MOS (↑)	DTW-MCD (↓)
Train and Test on LJSpeech		
Tacotron	3.27 ± 0.10	5.88 ± 0.13
Tacotron + Phoneme	3.31 ± 0.10	5.43 ± 0.10
Tacotron + HRG (ours)	$\mathbf{3.46 \pm 0.10}$	$\mathbf{4.79 \pm 0.08}$
TxTTS	3.33 ± 0.09	5.31 ± 0.12
TxTTS + HRG (ours)	$\mathbf{3.87 \pm 0.08}$	$\mathbf{4.70 \pm 0.10}$
Train and Test on Arctic		
Tacotron	1.84 ± 0.09	6.98 ± 0.17
Tacotron + Phoneme	2.93 ± 0.11	5.98 ± 0.07
Tacotron + HRG (ours)	$\mathbf{3.12 \pm 0.11}$	$\mathbf{5.21 \pm 0.09}$
TxTTS	1.80 ± 0.08	6.84 ± 0.13
TxTTS + HRG (ours)	$\mathbf{2.78 \pm 0.11}$	$\mathbf{5.88 \pm 0.12}$

Table 3. In-domain results: MOS scores and DTW-MCD values comparing speech quality of baselines: Tacotron, Tacotron + Phoneme, and TxTTS, with the proposed methods: Tacotron + HRG and TxTTS + HRG. We specify both the average value of the metric and the corresponding 95% confidence interval for the MOS and DTW-MCD scores.

Main Results (out-of-domain). In Table 4 we note the higher MOS scores and lower WER values for TxTTS + HRG, indicating that the listeners were better able to discern the words being spoken by the HRG model. The WER values are computed without any form of post-processing as noted in [26]. A substantial increase of 30% in MOS and a drop of 44% in WER clearly exhibits the ability of HRG based models to improve out-of-domain generalization. We observe inferior out-of-domain performance for Tacotron based models compared to TxTTS; hence we only experiment with the latter.

Train on LJSpeech and Test on Blizzard Conv [MOS (↑)]	
TxTTS	2.67 ± 0.07
TxTTS + HRG (ours)	$\mathbf{3.49 \pm 0.09}$
Train on LJSpeech and Test on Blizzard SUS [WER (↓)]	
TxTTS	40.76%
TxTTS + HRG (ours)	**22.67%**

Table 4. Out-of-domain results: MOS scores on Blizzard Conv and WER metrics on Blizzard SUS, comparing the speech quality of TxTTS and TxTTS + HRG.

[5] https://github.com/festvox/festvox/blob/master/src/clustergen/get_cd_dtw.

[6] For transparency, we also provide the outputs used for human evaluation at https://tinyurl.com/5ebpdnra.

In comparison to other graph-based methods. Our results which highlight the usefulness of a structural bias inherent in the HRGs are not surprising since contemporary works like [8, 9] have also investigated the benefits of syntactic information in the form of graph-based inputs and arrived at a similar conclusion. However, as we mention in Section 2, even with a more complicated approach, their graphs are rigid and less general than HRGs. We tried to re-implement the GraphTTS method [8], but we could not establish a good baseline. In the absence of a public implementation for [8, 9], we refrain from comparing our approach to their methods directly. We believe that our generic approach envelops the specific cases in each of them. Moreover, we make our implementation publicly available for ease of use.

Remark on MOS scores. The absolute values of the MOS scores on the Arctic datasets in Table 3 cannot be directly compared to scores in [3] (Table 2) and [5] (Table 1) since the Arctic dataset is much more challenging with its multiple male/female non-native speakers, each having only $\sim$ 1hr of sampled audio. In contrast, the internal dataset used by [3] consists of $\sim$ 24.6 hours of speech sampled from a single female native English speaker. Since the LJSpeech dataset has more hours of speech from a single speaker, the performance we observe on it is comparable to the MOS in [3] (Table 1). Furthermore, as MOS scores are derived from human judgements, the scores for the same model from two different experimental setups cannot be compared directly. This is because the absolute values of MOS are subject to annotator-specific biases [27]. Mayo et al. [28] used multidimensional scaling for identifying the main acoustic dimensions to which listeners attend when rating synthetic speech. They determine that several perceptually salient prosodic, segmental, and unit-level cues cause the listeners to undergo complex psychoacoustic processes influencing their decisions on the naturalness of the generated speech. Due to similar reasons, benchmarks like the Blizzard challenge [16] use a single standard dataset. Hence, because of the unavailability of standardized human evaluation sets for LJSpeech and Arctic, we limit our discussion to the relative performance improvements in the in-domain setting.

Analysis. To understand the performance improvements observed for the HRG model, we evaluate the utility of HRGs to better predict post-alignment acoustic features like phoneme level durations. For each phoneme node in the HRG, the objective is to predict a discretized value of the *duration*, which captures the time needed to pronounce it. We divide the durations into ten discrete buckets (classes) and train two models for duration classification: **i)** HRG: An MLP classification head is attached to our GCN module. Essentially, this method uses the phoneme representations learned by our GCN module to perform duration prediction (Section 3); and **ii)** BiLSTM: As a strong baseline, we train a bi-directional LSTM [29] based sequence-to-sequence model [30] with global attention [31] and a hidden size of 500. Given an input sequence of phonemes, the BiLSTM based method generates a sequence of duration nodes (one for each phoneme) as the output. Both the models were trained for 10 epochs and we report the test accuracies of best validation checkpoints in Table 5.

Model	Accuracy
BiLSTM	24.72
HRG (ours)	**71.89**

Table 5. Analysis for the task of duration prediction. Comparing the accuracies of the baseline BiLSTM model with our GCN based HRG model.

The HRG based encoder clearly outperforms the BiLSTM baseline in classifying phonemes into discretized durations. We believe that these gains can be attributed to the fact that the duration of a phoneme node depends predominantly on the local context of neighboring phonemes and syllables. This dependence is efficiently captured by the localized GCN convolutions over the HRG nodes. The GCN representations drawn from structural features (like the word to phoneme hierarchies or syllable boundaries present in HRGs) enable our approach to naturally learn these dependencies and invariances.

6. CONCLUSION

In this work, we take the first steps to re-purpose Heterogeneous Relation Graphs (HRGs) proposed by Taylor et al. [13] to deterministically encode linguistic structure present in a text sequence. We propose to feed the HRGs consisting of syllables and phonemes as inputs to popular end-to-end neural TTS models like Tacotron and TransformerTTS. Graph Convolution Networks are used to learn continuous representations for each phoneme node in the HRG, which are passed as a stream of vectored inputs to the base neural TTS model.

As indicated by MCD and MOS scores, this simple adaptation furnishes significant improvements in speech quality on both single speaker, large sample dataset LJSpeech; and multi-speaker dataset Arctic which has much fewer training samples per speaker. Additionally, we see that when the TransformerTTS model is trained using HRGs, the out-of-domain performance also improves. This is confirmed by higher MOS and lower WER scores (on the 2005 Blizzard Challenge test sets) for TransformerTTS + HRG trained on LJSpeech corpus. Finally, we conclude with an empirical analysis that demonstrates the ability of HRG trained models to better predict post-alignment acoustic features like phoneme durations. In the future, we plan to train end-to-end TTS models that implicitly learn the relationships between phonemes, syllables and other linguistic units in the natural text input. In such a scenario, HRGs would serve more as an implicit regularizer as opposed to an explicit input.

References

[1] A. J. Hunt and A. W. Black, “Unit selection in a concatenative speech synthesis system using a large speech database,” in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1. IEEE, 1996, pp. 373–376.

[2] H. Zen, K. Tokuda, and A. W. Black, “Statistical parametric speech synthesis,” *speech communication*, vol. 51, no. 11, pp. 1039–1064, 2009.

[3] Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio *et al.*, “Tacotron: Towards end-to-end speech synthesis,” *arXiv preprint arXiv:1703.10135*, 2017.

[4] J. Shen, R. Pang, R. J. Weiss, M. Schuster, N. Jaitly, Z. Yang, Z. Chen, Y. Zhang, Y. Wang, R. Skerrv-Ryan *et al.*, “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2018, pp. 4779–4783.

[5] N. Li, S. Liu, Y. Liu, S. Zhao, and M. Liu, “Neural speech synthesis with transformer network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 6706–6713.

[6] G. K. Anumanchipalli, K. Prahallad, and A. W. Black, “Festvox: Tools for creation and analyses of large speech corpora,” in *Workshop on Very Large Scale Phonetics Research, UPenn, Philadelphia*, 2011, p. 70.

[7] G. Zhang, Y. Qin, and T. Lee, “Learning syllable-level discrete prosodic representation for expressive speech generation,” *Proc. Interspeech 2020*, pp. 3426–3430, 2020.

[8] A. Sun, J. Wang, N. Cheng, H. Peng, Z. Zeng, and J. Xiao, “Graphtts: graph-to-sequence modelling in neural text-to-speech,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2020, pp. 6719–6723.

[9] R. Liu, B. Sisman, and H. Li, “Graphspeech: Syntax-aware graph attention network for neural speech synthesis,” in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2021, pp. 6059–6063.

[10] A. Setlur, B. Póczos, and A. W. Black, “Nonlinear isa with auxiliary variables for learning speech representations,” *arXiv preprint arXiv:2007.12948*, 2020.

[11] A. v. d. Oord, O. Vinyals, and K. Kavukcuoglu, “Neural discrete representation learning,” *arXiv preprint arXiv:1711.00937*, 2017.

[12] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, “Unsupervised speech representation learning using wavenet autoencoders,” *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 12, pp. 2041–2053, 2019.

[13] P. Taylor, A. W. Black, and R. Caley, “Heterogeneous relation graphs as a formalism for representing linguistic information,” *Speech Communication*, vol. 33, no. 1-2, pp. 153–174, 2001.

[14] A. Black and K. Lenzo, “Building voices in the festival speech synthesis system,” 2000.

[15] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.

[16] A. W. Black and K. Tokuda, “The blizzard challenge-2005: Evaluating corpus-based speech synthesis on common datasets,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[17] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, “Wavenet: A generative model for raw audio,” in *9th ISCA Speech Synthesis Workshop*, 2016, pp. 125–125.

[18] D. Griffin and J. Lim, “Signal estimation from modified short-time fourier transform,” *IEEE Transactions on acoustics, speech, and signal processing*, vol. 32, no. 2, pp. 236–243, 1984.

[19] R. Prenger, R. Valle, and B. Catanzaro, “Waveglow: A flow-based generative network for speech synthesis,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 3617–3621.

[20] S. Ronanki, O. Watts, and S. King, “A hierarchical encoder-decoder model for statistical parametric speech synthesis.” in *INTERSPEECH*, 2017, pp. 1133–1137.

[21] A. Sun, J. Wang, N. Cheng, H. Peng, Z. Zeng, L. Kong, and J. Xiao, “Graphpb: Graphical representations of prosody boundary in speech synthesis,” *arXiv preprint arXiv:2012.02626*, 2020.

[22] K. Mametani, T. Kato, and S. Yamamoto, “Investigating context features hidden in end-to-end tts,” in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* IEEE, 2019, pp. 6920–6924.

[23] K. Ito, “The lj speech dataset,” https://keithito.com/LJ-Speech-Dataset/, 2017.

[24] A. W. Black, “Clustergen: A statistical parametric synthesizer using trajectory modeling,” in *Ninth International Conference on Spoken Language Processing*, 2006.

[25] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[26] C. L. Bennett, “Large scale evaluation of corpus-based synthesizers: Results and lessons from the blizzard challenge 2005,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[27] S. Zielinski, F. Rumsey, and S. Bech, “On some biases encountered in modern audio quality listening tests-a review,” *Journal of the Audio Engineering Society*, vol. 56, no. 6, pp. 427–451, 2008.

[28] C. Mayo, R. A. Clark, and S. King, “Multidimensional scaling of listener responses to synthetic speech,” in *Ninth European Conference on Speech Communication and Technology*, 2005.

[29] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[30] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” in *3rd International Conference on Learning Representations, ICLR 2015*, 2015.

[31] M.-T. Luong, H. Pham, and C. D. Manning, “Effective approaches to attention-based neural machine translation,” in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1412–1421.