МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**
Кафедра математической кибернетики и компьютерных наук

ПЕРЕНОС СТИЛЯ ЧЕЛОВЕЧЕСКОГО ГОЛОСА С ИСПОЛЬЗОВАНИЕМ НЕЙРОННЫХ СЕТЕЙ

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Толобаева Игоря Олеговича

Научный руководитель
доцент, к. ф.-м. н. _____ А. С. Иванова

Заведующий кафедрой
к. ф.-м. н., доцент _____ А. С. Иванов

Саратов 2020

ВВЕДЕНИЕ

Идея говорить чьим-то голосом не перестает быть увлекательным элементом сценариев боевиков и фантастических фильмов. Также она также находит свое применение во многих практических приложениях, таких как защита конфиденциальности и идентичности, творческая индустрия и т.д. Такая задача часто называется задачей переноса стиля на голосе и включает в себя модификацию заданной речи исходного говорящего, чтобы соответствовать речевым особенностям целевого говорящего.

Несмотря на продолжающиеся исследования в области преобразования голоса, три проблемы остаются недостаточно изученными. Во-первых, большинство систем преобразования голоса предполагают наличие параллельных обучающих данных, т.е. речевых пар, где два говорящих произносят одинаковые предложения. Лишь некоторые модели могут обучаться на непараллельных данных. Во-вторых, среди подобных немногих алгоритмов еще меньшее количество может работать с преобразованием вида «многие ко многим», т.е. преобразованием от нескольких исходных говорящих к нескольким целевым говорящим. Наконец, ни одна предшествующая система преобразования голоса не может выполнить преобразование с использованием zero-Shot обучения, т.е. преобразование в голос неизвестного говорящего, взглянув лишь на несколько его/ее высказываний.

С недавними достижениями в области использования глубокого обучения для переноса стиля, традиционная проблема преобразования голоса превращается в проблему переноса стиля, где речевые особенности можно рассматривать как стиль, а говорящих — как домены. Существует несколько алгоритмов переноса стиля, которые не требуют параллельных данных и применимы к нескольким доменам, поэтому они легко доступны в качестве решений для преобразования голоса. В частности, в преобразовании голоса набирают популярность генеративные состязательные сети (GAN) и условный вариационный автокодировщик (CVAE).

Однако ни GAN, ни CVAE не идеальны. GAN имеет хорошее теоретическое обоснование того, что сгенерированные данные будут соответствовать распределению истинных данных. GAN достигает самых впечатляющих результатов, особенно в области компьютерного зрения. Тем не менее, общепризнанно, что GAN очень трудно обучить, и его свойство сходимости очень

нестабильно. Кроме того, хотя появляется все большее число работ, посвященных созданию GAN для генерации речи и передачи стиля речи, нет убедительных доказательств того, что сгенерированная речь ***звучит*** правдоподобно. Речь, которая способна обмануть машину, еще не может обмануть человеческие уши. CVAE, с другой стороны, легче обучать, однако результат часто страдает от чрезмерного сглаживания выходных данных преобразования. [1]

В 2019 году была предложена новая схема переноса стиля под названием AutoVC, в которой используется только автокодировщик. Подобно CVAE, предложенная схема обучается с использованием только функции потери от перепостроения, но одновременно с этим модель имеет свойство соответствия распределения, также как и GAN. Это связано с тем, что правильно разработанное узкое место обучается удалять информацию о стиле из источника и получать некоторый вектор, не зависящий от стиля, что и является целью CVAE, но которую схема обучения CVAE не может гарантировать.

Эта простая схема приводит к значительному увеличению качества результата, обеспечивая приемлемый результат при выполнении традиционной задачи преобразования вида «многие ко многим», когда все говорящие присутствуют в обучающем наборе. [2] В данной архитектуре используется векторное представления говорящего, обученное для определения говорящего, что позволяет использовать эффективное zero-Shot обучение. Учитывая качество результатов и простоту схемы обучения, попытки улучшения данной архитектуры могут открыть путь к более простым и качественным системам преобразования голоса и переноса стиля.

Целью выпускной квалификационной работы (ВКР) является построение системы переноса стиля с использованием нейронных сетей.

Поставленная цель определяет следующие задачи:

— Разработать архитектуру переноса стиля на голосе с использованием существующих нейронных сетей;
— Создать программу, осуществляющую кодирование говорящего с использованием нейронной сети;
— Обучить нейронную сеть для переноса стиля на голосе;
— Создать программу, осуществляющую построение звукового файла на основе mel-спектрограммы.

1 Архитектура системы переноса стиля

Нейронная сеть состоит из трех основных модулей: кодировщик говорящего, кодировщик содержания, декодер. На вход подается mel-спектрограмма с речью размера N на T, где N — разрешение mel-спектрограммы, а T — количество временных шагов (кадров). Инвертор спектрограммы нужен, чтобы преобразовать выходную спектрограмму обратно в звуковую волну, что также будет подробно описано в этом разделе.

1.1 Автокодировщик переноса стиля

Проблема преобразования голоса может быть решена с помощью очень простой структуры автокодировщика. Он состоит из трех модулей: кодировщика содержания $E_c(\cdot)$, который создает вектор содержания речи, кодировщика говорящих $E_s(\cdot)$, который позволяет получить вектор говорящего из его речи, и декодера $D(\cdot, \cdot)$, который создает речь из векторов содержания и говорящих. Входные данные для этих модулей различны для преобразования и обучения.

Преобразование: Во время фактического преобразования исходная речь X_1 подается на кодировщик содержания для извлечения информации. Целевая речь подается на кодировщик говорящего для получения информации о целевом говорящем. Декодер создает преобразованную речь на основе информации о содержании в исходной речи и информации о говорящем в целевой речи.

$$C_1 = E_c(X_1), \quad S_2 = E_s(X_2), \quad \hat{X}_{1\to 2} = D(C_1, S_2). \tag{1}$$

Здесь C_1 и $\hat{X}_{1\to 2}$ являются случайными процессами. S_2 — это просто случайный вектор.

Обучение: Поскольку не предполагается наличие параллельных данных, для обучения необходимо только перепостроение исходных данных. То есть вход для кодировщика содержания по-прежнему составляет X_1, а входом для кодировщика стиля становится высказывание того же говорящего U_1, обозначенное как X_1'. При этом X_1' и X_1 могут быть как одинаковыми, так и различными. Затем каждое входное сообщение X_1 учится производить построение самого себя:

$$C_1 = E_c(X_1), \quad S_1 = E_s(X_1'), \quad \hat{X}_{1\to 1} = D(C_1, S_1). \tag{2}$$

Функция минимизации потерь — это просто взвешенная комбинация ошибки построения высказывания и ошибки построения содержания, т.е.

$$\min_{E_c(\cdot),D(\cdot,\cdot)} L = L_{\text{recon}} + \lambda L_{\text{content}}, \tag{3}$$

где

$$\begin{aligned} L_{\text{recon}} &= \mathbb{E}[\|\hat{X}_{1\to1} - X_1\|_2^2], \\ L_{\text{content}} &= \mathbb{E}[\|E_c(\hat{X}_{1\to1}) - C_1\|_1]. \end{aligned} \tag{4}$$

Как оказалось, этой простой схемы обучения достаточно для создания идеального преобразования голоса, соответствующего распределению.

1.2 Промежуточное представление признаков

Использование представления, которое легко вычисляется из звуковых сигналов во временной области, позволяет обучать два компонента по отдельности. В данном случае таким представлением будет mel-спектрограмма. Это представление также более плавное, чем звуковые семплы, и его легче обучить с использованием квадратичной функции потерь, поскольку оно инвариантно к фазе в каждом кадре. [3]

В то время, как линейные спектрограммы отбрасывают информацию о фазе (и, следовательно, являются преобразованием с потерями), такие алгоритмы, как алгоритм Гриффина-Лима, способны оценивать эту отброшенную информацию, что обеспечивает преобразование во временной области посредством обратного кратковременного преобразования Фурье. [4]

Mel-спектрограммы отбрасывают еще больше информации, усложняя задачу обратного преобразования. Однако по сравнению с лингвистическими и акустическими признаками, используемыми в WaveNet, mel-спектрограмма представляет собой более простое и низкоуровневое акустическое представление аудиосигналов. Следовательно, для аналогичной WaveNet модели, основанной на спектрограммах, должно быть просто генерировать звук в виде нейронного вокодера. Кроме того, используя 80 частотных сегментов для вычисления спектрограммы с интервалом кадров 12,5 мс, для представления каждого кадра потребовалось бы только 80 значений по сравнению с 300 выборками в форме звуковой волны с частотой 24 кГц, что делает предсказание более легким.

1.3 Кодировщик говорящего

Кодировщик говорящего используется для настройки сети обработки под требуемого целевого говорящего. Критическим для возможности обобщения является использование представления, которое фиксирует характеристики разных говорящих, и возможность идентифицировать эти характеристики, используя только короткий сигнал, независимо от его фонетического содержания и фонового шума. Эти требования выполняются с использованием дискриминационной модели говорящего, обученной на задаче определения говорящего, независимо от текста.

Цель кодировщика говорящего состоит в том, чтобы получать одинаковое векторное представление для разных высказываний одного и того же говорящего и разные векторные представления для разных говорящих. Для обычного преобразования голоса типа «многое ко многому» достаточно one-hot кодирования векторов говорящих. Однако, чтобы выполнить zero-shot преобразование, нужно получить вектор, который может быть распространен на недоступных говорящих. Кодировщик говорящего состоит из стека из двух слоев LSTM по 768 нейронов. Берется только последний по времени выход, а размерность понижается до 256 с помощью полносвязного слоя. Результат таким образом представляет собой вектор 256 на 1. Кодировщик говорящих предварительно обучен работе с использованием функции потерь GE2E (версия softmax), которая максимизирует сходство векторов между различными высказываниями одного и того же говорящего и сводит к минимуму сходство среди разных говорящих. [5]

1.4 Кодировщик содержания

Вход для кодировщика содержания — это 80-мерная mel-спектрограмма X_1, объединенная с вектором говорящего $E_s(X_1)$ на каждом временном шаге. Объединенные элементы подаются на три сверточных 5×1 слоя, каждый из которых сопровождается батч-нормализацией и функцией активацией ReLU. Количество каналов составляет 512. Выходные данные затем передаются в стек из двух двунаправленных слоев LSTM. Размеры прямых и обратных ячеек равны 32, поэтому их объединенная размерность равна 64.

В качестве ключевого шага построения узкого места, прямой и обратный выходы двунаправленного LSTM понижаются до 32. Понижение частоты

выполняется по-разному для прямого и обратного направлений. Для прямого вывода временные шаги составляют $\{0, 32, 64, \cdots\}$; для обратного вывода сохраняются временные шаги составляют $\{31, 63, 95, \cdots\}$. Результирующий вектор содержания представляет собой набор из двух матриц 32 на $T/32$.

1.5 Декодер

Tacotron представляет собой архитектуру типа последовательность-последовательность для создания спектрограмм из последовательности символов, что упрощает традиционный конвейер синтеза речи, заменяя получение этих лингвистических и акустических признаков единственной нейронной сетью, обученной только на основе данных. Чтобы воспроизвести результирующие спектрограммы, Tacotron использует алгоритм Гриффина-Лима для оценки фазы, за которым следует обратное преобразование Фурье.

Как было отмечено в [6], это является просто заготовкой для будущих подходов нейронных вокодеров, поскольку алгоритм Гриффина-Лима производит характерные артефакты и более низкое качество звука, чем подходы, подобные WaveNet.

Как и в Tacotron, будем вычислять спектрограммы с помощью кратковременного преобразования Фурье (STFT) с использованием размера кадра в 50 мс. Преобразуем величину STFT в спектрограмму, используя 80-канальный набор фильтров, охватывающий диапазон от 125 Гц до 7,6 кГц, с последующим сжатием динамического диапазона. Перед сжатием выходные величины набора фильтров обрезаются до минимального значения 0.01, чтобы ограничить динамический диапазон областью определения логарифма.

Сеть состоит из кодировщика и декодера. Кодировщик преобразует последовательность символов в представление скрытых признаков, которое декодер использует для предсказания спектрограммы. Входные символы представляют собой вектор размера 512, который пропускаются через стек из 3 сверточных слоев, каждый из которых содержит 512 фильтров с формой 5×1, то есть где каждый фильтр охватывает 5 символов, после чего следует батч-нормализация и функция активации ReLU. Как и в Tacotron, эти сверточные слои моделируют долгосрочный контекст т.е. N-граммы во входной последовательности символов. Выходной сигнал конечного сверточного слоя передается в один двунаправленный слой LSTM, содержащий 512 нейронов (256 в каждом направлении) для генерации закодированных признаков.

Выходные данные кодировщика используются сетью Attention, которая суммирует полную закодированную последовательность как контекстный вектор фиксированной длины для каждого шага декодера. Будем использовать Attention, что позволит использовать совокупные веса из предыдущих шагов декодера в качестве дополнительного признака. [7] Это побуждает модель последовательно продвигаться вперед по входным данным, смягчая возможные неудачи, когда некоторые подпоследовательности повторяются или игнорируются декодером. Вероятности Attention вычисляются после проецирования входных данных и признаков местоположения в 128-мерные скрытые представления. Признаки местоположения вычисляются с использованием 32-х одномерных сверточных фильтров длиной 31.

Декодер является авторегрессионной рекуррентной нейронной сетью, которая предсказывает спектрограмму на основе закодированной входной последовательности по одному кадру за раз. Предсказание из предыдущего временного шага сначала передается через небольшую сеть, содержащую 2 полносвязных слоя из 256 скрытых нейронов с функцией активации ReLU. Эта сеть, выступающая в качестве информационного узкого места, необходима для обучения Attention. Выходные данные сети и вектор контекста Attention объединяются и пропускаются через стек из 2 однонаправленных слоев LSTM с 1024 нейронами. Объединение выходных данных LSTM и вектора контекста Attention проецируется с помощью линейного преобразования для предсказания целевого кадра спектрограммы. Наконец, предсказанная спектрограмма пропускается через 5-слойную сверточную сеть, которая предсказывает остаток, суммируемый с предсказанием, чтобы улучшить общий результат. Каждый слой этой сети состоит из 512 фильтров размерности 5×1 с батч-нормализацией, за которыми следуют функции активации $\tanh$ на всех, кроме последнего слоя.

Параллельно с предсказанием кадра спектрограммы конкатенация выходных данных LSTM декодера и контекста Attention превращается в скаляр и проходит через сигмоидную функцию активацию, чтобы предсказать вероятность завершения выходной последовательности. Это предсказание «stop token» используется во время вывода, чтобы позволить модели динамически определять, когда прекратить генерацию, вместо того чтобы осуществлять генерацию в течение фиксированной продолжительности. В частности, гене-

рация завершается в первом кадре, для которого эта вероятность превышает порог 0.5.

Сверточные слои в сети упорядочены с использованием dropout с вероятностью 0.5, а слои LSTM упорядочены с использованием zoneout с вероятностью 0.1. Чтобы ввести изменение выходного сигнала во время вывода, к слоям в предварительной сети авторегрессионного декодера применяется dropout с вероятностью 0.5.

В отличие от оригинального Tacotron, данная модель использует более простые составные части, используя в кодировщике и декодере обычный LSTM и сверточные слои вместо стеков «CBHG» и рекуррентных слоев «GRU». «Коэффициент уменьшения» при этом использоваться не будет, то есть каждый шаг декодера будет соответствовать одному кадру спектрограммы.

Чтобы лучше построить тонкие детали спектрограммы поверх начальной оценки, вводится сеть пост-обработки. [8] Эта сеть состоит из пяти 5×1 сверточных слоев, где к первым четырем слоям применяются батч-нормализация и гиперболический тангенс. Размер канала для первых четырех слоев составляет 512, а в последнем слое — 80.

1.6 Обратная спектрограмма

В представленной нейронной сети для получение результирующего аудиосигнала применяется вокодер WaveNet, который состоит из четырех слоев обратной свертки. В данном случае частота кадров спектрограммы составляет 62,5 Гц, а частота дискретизации речевого сигнала составляет 16 кГц. Таким образом, слои обратной свертки будут дискретизировать спектрограмму, чтобы соответствовать частоте дискретизации речевого сигнала. Затем для генерации речевого сигнала применяется стандартная 40-слойная WaveNet на спектрограмме увеличенного размера. Вокодер WaveNet был предварительно обучен, на корпусе VCTK. [9]

Аналогично PixelCNNs, условное распределение вероятностей моделируется стеком сверточных слоев. В сети нет pooling слоев, а выходные данные модели имеют такую же временную размерность, что и входные. Поскольку логарифмическое правдоподобие поддаются анализу, имеется возможность тюнить гиперпараметры на валидационной выборке и легко контролировать переобучение. [10]

2 Обучение сети и осуществление переноса стиля

Реализация архитектуры сети для непосредственного переноса стиля, впервые описанная в [2], была выложена ее авторами на Github. К сожалению, предоставленного кода достаточно, лишь чтобы проверить работоспособность уже обученной модели. Предполагается, что будут использоваться заранее предобученные модели AutoVC (для переноса стиля) и WaveNet Vocoder (для превращения спектрограммы в звуковую волну). Любой код, который бы позволял обучить собственные модели был намеренно убран из репозитория. Более того, в репозиторий помещен файл *metadata.pkl*, который представляет собой входные данные модели преобразования. Данный файл содержит тройки типа *(название, вектор говорящего, спектрограмма)*. Данные тройки являются результатом работы кодировщика говорящего, который также не представлен в репозитории. В представленном файле содержится информация о четырех говорящих из датасета VCTK. Общий же размер датасета составляет 109 носителей различных акцентов английского языка, где каждый из говорящих читает примерно 400 предложений. Как упоминалось выше, из 109 говорящих в свободный доступ были выложены векторы только четырех. Более того, из 400 предложений на каждого из говорящих была оставлена спектрограмма только одного.

В [5] описывается создание вектора говорящего на основе спектрограмм звуковых файлов с его речью. Размерность созданного таким образом результирующего вектора совпадает с размерностью вектора, который сеть переноса стиля принимает на вход. Это позволило при выполнении ВКР создать собственный файл, аналогичный *metadata.pkl* с векторами произвольных говорящих.

2.1 Кодировщик содержания

В ходе выполнения ВКР был написан скрипт, который может использовать обученную модель для получения тройки *(название, вектор говорящего, спектрограмма)*. Названием в данном случае будет название директории каждого говорящего. В датасете VCTK названия этих директорий имеют вид «pXXX», где XXX — номер говорящего. Спектрограмма же может быть получена с использованием библиотеки librosa, где реализована нужная функция. Параметр разрешающей способности спектрограмм установим на 80, как было

указано в оригинальной работе.

2.2 Цикл обучения

Когда входные данные для модели созданы, модель может быть обучена. Во время выполнения ВКР был разработан скрипт ее обучения, используя уже реализованную архитектуру модели. Функция потерь будет представлять собой взвешенную сумму трех компонентов: компонент потери начального приближения (content0 loss), компонент потери после применения postnet (content loss) и компонент потери говорящего (speaker loss).

Так как сеть используется в режиме тренировки, в генератор кроме спектрограммы два раза передается вектор говорящего. На выходе получаем две спектрограммы. Первая из них представляет собой результат воссоздания спектрограммы основной сетью и будет участвовать в подсчете content0 loss. Вторая же является результатом последующего применения дополнительной сети postnet и будет участвовать в подсчете content loss.

На этом этапе остается лишь посчитать все три компонента функции потерь. Так как спектрограмма по своей сути представляет собой изображение, то есть двумерный массив, мы можем использовать среднее квадратическое отклонение в качестве функции потерь. В качестве функции потерь для вектора говорящего используем метод наименьших модулей (L1 Loss). Так как полученное значение функции потерь на несколько порядков меньше значений предыдущих двух функций, умножим значение на коэффициент 60000. За счет этого во время обучения модели значение третьего компонента функции потерь будет считаться примерно таким же важным, как и первые два.

После обучения модель может быть использована для генерации результирующей спектрограммы. В режиме преобразования в генератор передается исходная спектрограмма и два вектора: вектор исходного говорящего и вектор целевого говорящего.

2.3 Вокодер

Последним шагом является преобразование полученной спектрограммы в звуковую волну. Так как преобразование звуковой волны в спектрограмму — это преобразование с потерями, восстановление оригинальной звуковой волны по спектрограмме — задача не тривиальная. Существует множество отличающихся друг от друга звуковых файлов, полученных из одной спектрограммы.

Более того, звуковые файлы, полученные из спектрограмм относительно простыми методами обычно страдают от потери качества при преобразовании. Очевидно, что у системы переноса стиля на голосе нет ценности, в случае если результирующий аудиофайл неудовлетворительного качества, поэтому решение данной проблемы чрезвычайно важно.

Для решение этой задачи в ходе выполнения ВКР был использован вокодер на основе нейронной сети. Подобный вокодер, который принимает на вход спектрограмму и превращает ее в звуковую волну описывается как часть сети WaveNet и изначально использовался для задачи Text to Speech.

Тогда, единственное, что остается — это загрузить предварительно обученный вокодер и подать ему на вход полученную на предыдущем шаге спектрограмму.

2.4 Эксперименты

В качестве входных данных zero-shot преобразования уже обученной модели при выполнении ВКР были использованы фрагменты аудиокниги «The Things They Carried», озвученной голливудским актером Брайаном Крэнстоном. Аудиокнига была разделена на фрагменты длиной 4 секунды. Один из таких фрагментов будет использован в качестве аудио целевого говорящего. В качестве же исходного говорящего будем использовать один из женских голосов датасета VCTK.

Для того, чтобы использовать данные фрагменты для преобразования, создадим векторное представление говорящих с помощью кодировщика, а также преобразуем исходные аудиофайлы в спектрограммы.

В качестве сети переноса стиля была использована модель, обученную ранее на датасете VCTK. В итоге получим результирующую спектрограмму с голосом Брайана Крэнстона, который произносит фразу, сказанную говорящим из датасета VCTK.

ЗАКЛЮЧЕНИЕ

В ходе выполнения ВКР была создана система переноса стиля на голосе на основе современного алгоритма непараллельного преобразования.

При выполнении ВКР:

- создана программа, осуществляющая кодирование говорящего с использованием нейронной сети;
- обучена нейронная сеть для непосредственного переноса стиля голоса;
- разработана программа, осуществляющая построение результирующего звукового файла на основе mel-спектрограммы.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1 Kameoka, H., Kaneko, T., Tanaka, K., and Hojo, N. Stargan-vc: Non-parallel many-to-many voice conversion with star generative adversarial networks. *arXiv preprint arXiv:1806.02169*, 2018.

2 Qian, K., Zhang Y., Chang, S., Yang X., Hasegawa-Johnson, M. AUTOVC: Zero-Shot Voice Style Transfer with Only Autoencoder Loss. In *ICML 2019*, 2019.

3 S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 4, pp. 357 – 366, 1980.

4 D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Transactions on Acoustics, Speech and Signal Processing*, pp. 236–243, 1984.

5 Wan, L., Wang, Q., Papir, A., and Moreno, I. L. Generalized end-to-end loss for speaker verification. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4879–4883. IEEE, 2018.

6 Y. Wang, R. Skerry-Ryan, D. Stanton, Y. Wu, R. J. Weiss, N. Jaitly, Z. Yang, Y. Xiao, Z. Chen, S. Bengio, Q. Le, Y. Agiomyrgiannakis, R. Clark, and R. A. Saurous, "Tacotron: Towards end-to-end speech synthesis," in *Proc. Interspeech*, Aug. 2017, pp. 4006–4010.

7 J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015, pp. 577–585.

8 Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, RJ Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui. Wu. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2018.

9 van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., Kavukcuoglu, K., "WaveNet: A generative model for raw audio," *CoRR*, vol. abs/1609.03499, 2016.

10 van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.