电 子 科 技 大 学

UNIVERSITY OF ELECTRONIC SCIENCE AND TECHNOLOGY OF CHINA

学士学位论文

BACHELOR THESIS

论文题目 基于深度学习的四川话语音合成模型设计

学科专业 通信工程

学　　号 2019011209001

作者姓名 袁嘉程

指导老师 史创　副教授

摘　要

在本课题中，我们探索了基于对抗性训练的端到端文本到语音的变异推理（Variational Inference with adversarial learning for end-to-end Text-to-Speech,VITS）模型在语音合成中的表现。我们还评估了该模型在汉语和四川方言上的效果，期望取得良好的训练效果。VITS 是一个高表现力的语音合成模型，它结合了变分推理（variational inference）、归一化流（normalizing flows）和对抗性训练。在语音合成中，VITS 没有将声学模型和声码器串联起来，而是使用隐变量和随机持续时间预测器（stochastic duration predictor，SDP）来模拟语音的随机性，从而使同一输入文本的合成语音具有不同的音调和节奏。

我们的目标是建立中文和四川话的数据集和相应的预训练模块，可以转换中文符号。我们还将调整配置文件，特别是批量 (batch) 大小和历时 (epoch) 大小，并在评价指标方面将 VITS 模型与其他参考模型进行比较，如梅尔倒谱畸变（Mel-cepstral distortion，MCD），它客观地衡量两组 Mel-cepstral 系数的距离。我们将从测试集上的合成语音和相应的真实语音中提取 25 维的梅尔倒谱系数，帧移 5ms，以比较合成语音与原始音频之间的相似程度，以说明 VITS 模型的强大性能。我们还使用了平均意见得分（Mean Opinion Score，MOS），一个语音质量的主观评价标准，来评价合成语音的自然度和流畅度。

关键词：语音合成，深度学习，自然语言处理，声音信号处理，生成对抗网络

ABSTRACT

In this project, we explore the performance of an adversarial training-based Variational Inference with adversarial learning for end-to-end Text-to-Speech (VITS) model in speech synthesis. We also evaluate the model on Chinese and Sichuan dialects, expecting good training results.VITS is a highly expressive speech synthesis model that combines variational inference, normalizing flows, and adversarial training. In speech synthesis, instead of concatenating acoustic models and vocoders, VITS uses hidden variables and stochastic duration predictor (SDP) to model the randomness of speech, resulting in synthetic speech with different pitches and rhythms for the same input text. Although English models based on the LJspeech dataset have been proposed and tested in previous studies, we aim to build Chinese and Sichuanese(a local Chinese dialect) datasets and corresponding pre-training modules that can transform Chinese symbols. We will also adjust the profiles, especially the batch size and epoch size, and compare the VITS model with other reference models in terms of evaluation metrics, such as Mel-cepstral distortion (MCD), which objectively measures the distance between two sets of Mel- cepstral coefficients at a distance. We will extract 25-dimensional Mel- cepstral coefficients from the synthesized speech and the corresponding real speech on the test set with a 5ms frame shift to compare the reconstruction performance of the synthesized speech with the original audio. We will also use Mean Opinion Score (MOS), a subjective evaluation criterion for speech quality to evaluate the naturalness of the synthesized speech and fluency of the synthesized speech.

Keywords: Speech Synthesis,Deep learning,Natural language Processing,Audio Signal Processing,Generative Adversarial Network

目 录

第一章 绪 论

1.1 研究工作的背景与意义

自然语言处理（Natural Language Processing，NLP）目前比较火热的研究方向，而使用深度学习进行的语音合成是 NLP 非常重要的一个里程碑。

语音合成技术 (Text To Speech，TTS)，是一种将文字域的信息转换到语音域的一种技术。语音合成技术经历了传统非端到端模型到现在流行的端到端语音合成的迭代。传统的语音合成系统是一种非端到端的，且包含两前端和后端两种模块组成的。前端模块主要对输入文本进行预处理和文本分析，以人工的方式，对文本中的一些语言学信息进行提取和规范化。完整的文本分析过程一般包含、词性预测、多音字消歧义，文本正则化、分词以及韵律预测五大任务，根据具体的语音合成语种不同，处理过程也会有所差异。比如英文合成，只需要文本正则化一步也可达到不错的效果。中文合成系统的前端部分一般包含所有五个任务。后端则通过一定方法，例如参数合成或拼接合成、生成语音波形。

而为了能生成出自然的，听得懂的，流利的合成语音，全球对这个课题进行了大量的研究。而深度学习的引入意味着语音合成从人工标记特征阶段走入神经网络时代，将构建好的数据集（训练样本）训练出比较好的语音合成模型是基于神经网络的语音合成的目标。目前，众多的算法和模型的引入，使得语音合成模型逐渐优化，合成效果越来越好。

基于变分推理和对抗性机械学习的端到端文语转换模型（Variational Inference with adversarial learning for end-to-end Text-to-Speech,VITS）[1] 是 2021 年由 Jaehyeon Kim 等人在论文中发表，其提出了一种平行的端到端 TTS 方法，与目前比较主流的两阶段模型相比，它可以单阶段的进行训练和并行，形成能产生更自然的音频。这篇文章使用了一些比较新颖的算法与方式，构建了一种对抗式的训练模型，实现了远超目前主流的流（Flow）[2]+ 生成对抗训练模型的语音合成效果。因此，对于 VITS 模型的研究，有利于了解目前语音合成领域的前沿技术，可以根据此进行更加深度的研究与优化。

本项目希望通过实现基于四川话的语音合成，自行构建 VITS 模型，提高模型对于中文的支持。体现 VITS 模型的优越之处。

1.2 TTS 模型研究现状及发展态势

TTS 模型的发展主要可以概括成非端到端模型到端到端模型的发展。非端到端 TTS 模型主要研究前端文本分析，声学模型和声码器三个部分的研究，而端到端模型则直接从文本音素转换到频谱或音频。由于前端文本分析需要利用语言学知识人工标注，往往在此过程中造成了信息缺失，构建模型的成本太高，因此，目前大多数研究都是基于端到端 TTS 模型的。目前，对 TTS 模型的研究主要针对快速语音合成（fast TTS）、低资源语音合成（low-resourceTTS）、鲁棒语音合成（robust TTS）、富有表现力的语音合成（expressive TTS）、可适配语音合成（adaptive TTS）等方面。本文所介绍的 VITS 模型便是一种完全端到端的，富有表现力的语音合成模型。以下对 4 种主要的 TTS 模型代表性技术分别进行描述。

1.Merlin

Merlin 是基于神经网络的语音合成。该系统将语言特征作为输入，采用神经网络来预测声学特征，然后将声学特征传递到声码器（Vocoder）以产生语音波形。Merlin 语音合成模型主要采用神经网络的架构，包括递归神经网络（Recurrent Neural Networks, RNN），长短时记忆网络（Long Short-Term Memory，LSTM）等等。

2. Tacotron&Tacotron2 [3](第一个真正意义上的端到端语音合成系统)

Tacotron 是第一个端对端的 TTS 神经网络模型，由 NVIDIA 在 2017 年发表，Tacotron 2 是一个可以直接从文本合成语音的神经网络模型。有两部分构成，第一部分是循环序列到序列的特征预测网络，其将字符嵌入层（Embedding，一种表征词语的序列）转换为梅尔频谱；第二部分是修改的非因果波形网络 (WaveNet [4]）模型，其作为语音合成器，将梅尔频谱合成为时域的波形进行预测。这两部分别进行训练。

3.FastSpeech&Fastspeech2 [5]

FastSpeech 是一个基于 Transformer 模型的端到端 TTS 模型，并采用了并行生成梅尔频谱的架构，最终实验结果表明 Fastspeech 比传统语音合成模型在梅尔频谱的生成上快 270 倍，在语音合成上快 38 倍，且不会对语音质量产生影响。其利用了知识蒸馏模型 [6](Knowledge Distillation)，使得浅层模型也能拥有和深层网络相近的性能。

虽然 fastspeech 已经取得了比传统自回归模型更快的语音合成速度和质量，但同样有一些缺点。比如（1）使用一个自回归的 TTS 模型作为教学（teacher）训练模型非常耗费时间；（2）使用知识蒸馏的方式来训练模型会导致信息损失，从而对合成出的语音的音质造成影响。而 FastSpeech2 针对这些缺点进行了改进，采用

了变量优化器[6]（variance adaptor）用来引入更多的输入来控制合成出的语音，从而在保持高质量合成语音的同时大幅度提高了合成速度。

4. WaveGAN[7]

WaveGAN是一个生成式对抗网络，用于无监督地合成原始波形音频，而不是时频图或语谱图。WaveGAN是通过生成器和判别器两部分组成；生成器上产生数据，如果判别模型能够成功判别，再修改参数产生新的数据，再判；而判别模型就是通过真实数据和模拟数据，判别准确率下去了，自动修改参数的两个相对独立过程构成的模型。

1.3 本文的主要贡献与创新

本论文在VITS模型基础上，设计了针对中文语音的文本预处理模型，使得中文文本能够规范化，以便之后进行相应的训练任务，获得较好的训练效果；

自制四川方言语音集，将训练迁移到对四川方言进行训练，获得了能够合成出表现力极强的四川方言语音合成模型，

设计了对比实验，使用tacotron2模型与VITS模型进行比较，在主观维度和客观指标上比较合成语音的表现力差异。

1.4 本论文的结构安排

本文的章节结构安排如下：在第二章中，论文介绍了传统语音合成模型的基础框架结构；第三章中，论文介绍了VITS模型的框架结构和所使用的算法，在原理上说明了VITS相对于传统算法的性能优势；在第四章中，论文详细的讲述了本文对VITS模型的研究内容，对设计的技术路线以及实验方案进行了详细的阐明；在第五章中，展示了我们设计的实验的结果，以证明VITS模型合成语音的高表现力；第六章中，我们对全文进行的总结，并对后续工作进行了展望。

第二章 传统语音合成模型架构

传统语音模型主要可以分为非端到端语音合成模型以及端到端语音合成模型。

2.1 非端到端语音合成模型

非端到端语音合成模型主要分为 4 个子模型：文本处理前端、时长模型、声学模型、声码器。

文本处理前端模块主要作用是将文本处理为特征向量，比如文字转音素、韵律标注、词性标注等。完整的前端文本处理过程一般包含文本正则化、分词、词性预测、注音以及韵律预测五大任务，根据具体的语音合成语种不同，处理过程也会有所差异。比如英文合成，只需要文本正则化一步也可达到不错的效果。一般来说，前端处理分为一下几个部分：

文本结构分析：通过判断输入文本的语言，调用相应的语言处理模块，来完成后续的文本处理任务。

文本正则：在非英文场景下，由于有一些英文或阿拉伯数字等非原始字符，因此需要把这一部分内容转化为原始语言的字符，以避免出现生成词语失去连贯性。比如在中文中，“疯狂星期四 v 我 50”，系统需要把“50”转化为“五十”，否则，生成的语音为连读的 5 和 0。

文本转音素：文本需要转化为归一化的计算机能够识别的字符，比如统一使用拼音或则罗马音。但是由于单一文本可能包含多种可能的音素表示，像中文中多音字的存在，所以如果直接随机的设置读音会影响生成语音的正确性，因此，我们需要找到能够正确翻译这些多义词的信息来辅助我们决策，包括了分词和每个词的词性。

韵律预测：通过决定一段话的节奏，也就是抑扬顿挫的程度，来预测生成音频的发音节奏。此所讲的韵律是从声学特征学习的具体表现形式，其内容可包含情感，语速，语音质量等级等等信息，主要使合成的语音更加自然，富有情感，对于每位说话人都是不同。

时长模型是根据文本特征向量预测每个向量的发音长度,效果是让任一文本都能与其对应语音对齐。

声学模型是根据文本的语言学特征以及长度信息，构建每个向量对应的声学特征，是语音合成中的关键步骤。

声码器（vocoder），又称语音信号分析合成系统，是根据声学模型得到的声学

特征（通常为梅尔谱或梅尔倒谱），反解码出最终得音频，最终实现语音的合成。

2.2 端到端语音合成模型

传统的非端到端语音合成系统对语言学，声学知识的要求较高。需要对不同语言的特点进行对应的特征提取，因此需要特定的语言学专家来辅助支持构建前端模块，同时，还需要有语音发声模块和信号处理模块的架构，增加了系统构建的难度，是相对复杂的系统。在语音数据集方面，这种非端到端的拼接系统则对语音数据库要求较高，对语音数据集的制作成本较大，费时费力。

这些问题促使端到端语音合成的出现，研究者希望能够语音合成系统能够尽量的简化，减少对人工干预和对语言学相关背景知识的依赖。因此，端到端语音合成系统由此诞生并大发展，成为了如今语音合成模型的主流架构。端到端语音合成模型，顾名思义，输入端是直接的文本而不是音素等人工提取的特征；而输出端是音频频谱或波形。在此基础上，前端模块和后端模块的结合性得到大大提高，前端模块得到极大简化。此外，端到端直接建立音频与文本之间的关系，因此各种可以很方便的进行不同语言间的迁移，而不需要人工对各种语言做特殊的标注处理，一个模型就可以合成多语种的语音。此外，端到端模型融入了深度学习的架构，结合诸多神经网络结构以及自然语言处理领域的相关算法，借助于这些模型的强表达能力，端到端语音合成系统可以对发音和韵律有非常好的建模，大大增强了其表现力。

端到端语音合成模型的架构一般可以由两部分组成：一部分用于预测文本的声学特征的声学模型，将字符嵌入层转化为梅尔频谱；另一部分声码器，将梅尔频谱转化为音频。

端到端语音合成系统的优点主要由一下几点：1）对提取语音的特征的工作量大幅度下降；2）更容易适应新的数据，比如不同语言，说话人之间的迁移；3）单个模型相对于组合的模型健壮性更强，在组合模型中，每个组件的错误都可能叠加而变得更加复杂。端到端语音合成模型的系统可以分为自回归模型和非自回归模型。

自回归模型不需要音素的对齐信息，而是在训练中自己学习音素的对齐关系，判断每一音素的发音时长，以及整个合成语音的时长；非自回归模型模型需要音素的对齐信息，需要其它工具提供音素对应的时长信息。

第三章 VITS 语音合成模型

3.1 模型结构原理

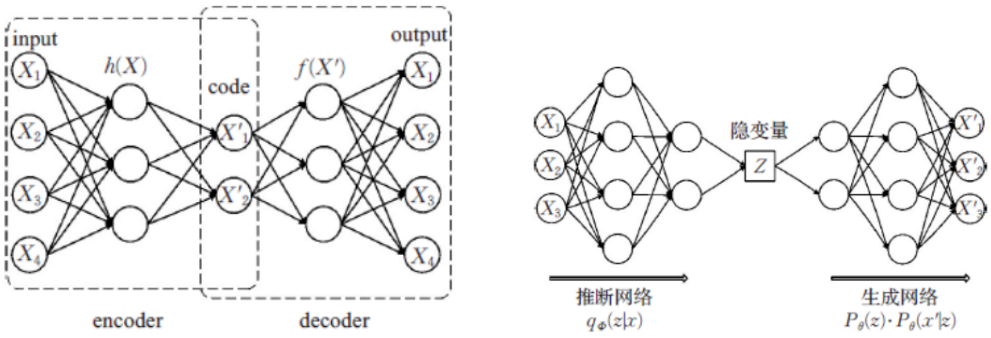
VITS 是一种结合变分推理[8]（variational inference）、标准化流[9]（normalizing flows）和生成对抗网络[10]（Generative Adversarial Networks，GAN）训练的高表现力语音合成模型。VITS 通过隐变量而非频谱串联起来语音合成中的声学模型和声码器，在隐变量上进行随机建模并利用随机时长预测器，提高了合成语音的多样性，输入同样的文本，能够合成不同声调和韵律的语音。其主要思想包括：条件变分自编码器[11]（Conditional Variational AutoEncoder，cVAE）、从变分推断中产生的对齐估计、生成对抗训练。

3.1.1 条件变分自编码器

变分自编码器[8]（Variational Auto-Encoders，VAE）作为深度生成模型的一种形式，是由 Kingma 等人于 2014 年提出的基于变分贝叶斯[12]（Variational Bayes，VB）推断的生成式网络结构。在 TTS 领域，我们可以采用 VAE 来获取语音的潜在空间变量，来对不同输入语音进行对潜在特征的映射。与传统的自编码器通过数值的方式描述潜在空间不同，在实际情况中，我们可能更多时候倾向于将每个潜在特征表示为可能值的范围。因此，VAE 以概率分布的方式描述对潜在空间的观察，利用潜在特征的概率分布来代替原先的单一数值来描述对特征的观察的模型。VAE 的提出对深度生成模型领域有着巨大的影响力，并和 GAN 模型一起被视为无监督式学习领域最具研究价值的方法之一。

VAE 模型类似于传统的 encoder-decoder 模型，利用两个神经网络建立两个概率密度分布模型，其中一个用于原始输入数据的变分推断，生成隐空间变量的概率分布，称为推断网络（inference network）；另一个根据生成的隐空间变量的概率分布，还原生成原始数据的近似概率分布，称为生成网络（generation network）。VAE 模型和传统的自编码器[13]（Auto-Encoders，AE）结构很类似，都由编码器（encoder）和解码器（decoder）构成，但其作用原理不同，VAE 的采用了概率分布来描述隐空间变量，而不是用文本嵌入层（word embedding）的向量表示。其结构如下图 2 所示。

VAE 模型通过将给定输入的每个潜在特征表示为概率分布。如果 X 表示原始数据样本的分布变量，产生的隐变量 Z 表示决定 X 特征的分布变量，该生成模型可以分成两个过程：



(a) (b)

图 3-1 VAE 模型结构

(1) 隐变量 Z 后验分布的近似推断过程：

$$P_\varphi(z|x) \tag{3-1}$$

即推断网络。

(2) 生成变量 X' 的条件分布生成过程：

$$P_\varphi(X') = P_\varphi(z)P_\varphi(x'|z) \tag{3-2}$$

即生成网络。

cVAE 在 VAE 的基础上，在训练时编码器和解码器部分都引入了数据标签，使其不仅可以重现输入，同时也能根据输入标签生成所需的输出数据。

VAE 在 TTS 模型中使用的基本原理是通过编码输入音频的梅尔频谱，得到相应的隐变量概率分布，并把这些分布当成语音的嵌入层来进行特定风格的语音合成。

3.1.2 变分推断

VITS 的生成器可以看作是一个最大化变分下界，也即证据下界 (Evidence Lower Bound，ELBO) 的条件 VAE。在训练过程中，VAE 会对样本进行采样，根据样本的不同信息，计算出相应的损失函数。

1. 重建损失

VITS 通过 flow 来完成线性谱输入的后验分布到先验分布之间的转换，增加模型表示能力。在训练时实际还是会生成梅尔频谱以指导模型的训练，重建损失中目标样本点使用的是梅尔频谱而非原始波形：在实现上，并不上采样整个隐变

量，而只是使用部分序列作为解码器的输入。

$$L_{recon} = ||x_{mel} - \hat{x}_{mel}|| \tag{3-3}$$

但在推断时不需要生成梅尔频谱，梅尔频谱仅仅是为了计算该重建损失。

2.KL 散度

KL 散度常常用于计算两个分布之间的差距。设一个概率分布 p 以及它的近似分布 p，则 KL 散度看的是对原始分布 p 中的数据概率与近似分布 q 之间的对数差的期望。即：

$$\begin{aligned} D_{KL}(p||q) &= \sum_{i=1} Np(x_i)(log\, p(x_i) - log\, q(x_i)) \\ &= E[log\, p(x_i) - log\, q(x_i)] \end{aligned} \tag{3-4}$$

利用 KL 散度，我们可以精确地计算出当我们近似一个分布与另一个分布时损失了多少信息。

对于一串可观测数据 x 和一系列隐变量 z，通过贝叶斯推理求解条件概率 $p(z|x)$, 需要使用变分推断。变分推断的目标是找到一个概率密度函数 $q(z)$ 来近似 $p(zx)$. 要得到最佳的 $q(z)$ 必须优化：

$$q^*(z) = \underset{q(z)\in Q}{argmax}\, KL(q(z)||p(z|x)) \tag{3-5}$$

其中，KL 散度可以表示为：

$$KL(q(z)||p(z|x)) = \mathbb{E}[log(q(z)] - \mathbb{E}[logp(z,x)] + logp(x) \tag{3-6}$$

由于 KL 散度大于 0，进而我们可以求得：

$$logp(x) \geq E[logp(z,x)]E[logq(z)] \tag{3-7}$$

上式中，右边部分为左边证据的对数形式，右边为其下界。即对于我们的目标原始分布 q(x)，其变分下界为

$$ELBO(q) = E[logp(z,x)]E[logq(z)] \tag{3-8}$$

在使用变分推断时，首先需要计算的便是 ELBO。

VITS 中，先验编码器 c 的输入包括 c_{text} 从文本生成的音素和音素、隐变量之间的对齐 A 。所谓的对齐就是 $|c_{text}| \times |z|$ 大小的严格单调注意力矩阵，表示每一个

音素的发音时长。因此 KL 散度是：

$$L_{kl} = log\, q_{\varphi}(z|x_{lin}) - log\, p_{\theta}(z|c_{text}, A) \tag{3-9}$$

其中，$q_{\varphi}(z|x)$ 表示给定线性谱 x 输出隐变量 z 的后验分布，$p_{\theta}(z|c)$ 表示给定条件 c 输出隐变量 z 的先验分布。其中隐变量 z 为：

$$z \sim q_{\varphi}(z|x_{lin}) = N(z; \mu_{\varphi}(x_{lin}), \sigma_{\varphi}(x_{lin})) \tag{3-10}$$

为了给后验编码器提供更高分辨率的信息，使用线性谱而非梅尔频谱作为后验编码器 q_{φ} 的输入。同时，为了生成更加逼真的样本，提高先验分布的表达能力比较重要，因此引入标准化流，以便在文本编码器产生的简单分布和隐变量 z 对应的复杂分布间进行可逆变换。也即在经过上采样的编码器输出之后，会加入如下的一系列可逆变换：

$$p_{\theta}(z|c) = N(f_{\theta}(z); \mu_{\theta}(c), \sigma_{\theta}(c))|det\frac{\partial f_{\theta}(z)}{\partial z}| \tag{3-11}$$

其中，输入 c 就是上采样的编码器输出：

$$c = [c_{text}, A] \tag{3-12}$$

3.1.3 对齐估计

由于在训练时没有”对齐“（alignment）的真实标签，因此在训练阶段的每一次迭代时都需要估计文本和音频之间的对齐。

1. 单调对齐搜索为了估计文本和语音之间的对齐 A，VITS 采用了类似于 Glow-TTS 中的单调对齐搜索[14]（Monotonic Alignment Search，MAS）方法，该方法试图寻找一个最优的对齐路径以最大化利用标准化流 f。其思想与 HMM 中使用的 Viterbi 很像。参数化数据的对数似然：

$$A = \underset{\hat{A}}{argmax}\, log\, p(x|c_{text}, \hat{A}) = \underset{\hat{A}}{argmax}\, N(f(x); \mu(c_{text}, \hat{A}), \sigma(c_{text}, \hat{A})) \tag{3-13}$$

MAS 约束获得的最优对齐必须是单调且无跳过的，但是无法直接将 MAS 直接应用到 VITS，因为 VITS 优化目标是 ELBO 而非确定的隐变量的对数似然，因此稍

微改变了一下 MAS，寻找最优的对齐路径以最大化 ELBO：

$$\begin{aligned} & argmax\,\hat{A}\,log\,p_\theta(x_{mel}|z) - log\,\frac{q_\varphi(z|x_{lin})}{p_\theta(z|c_{text},\hat{A})} \\ & = argmax\,\hat{A}\,log\,p_\theta(z|c_{text},\hat{A}) \\ & = log\,N(f_\theta(z);\mu_\theta(c_{text},\hat{A}),\sigma_\theta(c_{text},\hat{A})) \end{aligned} \tag{3-14}$$

2. 从文本中预测时长随机时长预测器[1]（stochastic duration predictor，SDP）是一个基于流的生成模型，引入与时长序列相同时间分辨率和维度的随机变量 u 和 v，利用近似后验分布 $q_\varphi(u,v|d,c_{text})$ 采样这两个变量，训练目标为音素时长对数似然的变分下界 ELBO：

$$log\,p_\theta(d|c_{text}) \geq \mathbb{E}_{q_\varphi(u,v|d,c_{text})}[log\,\frac{p_\theta(d-u,v|c_{text})}{q_\varphi(u,v|d,c_{text})}] \tag{3-15}$$

在训练时断开随机时长预测器的梯度反传，防止该部分梯度影响到其它模块。音素时长通过随机时长预测器的可逆变换从随机噪音中采样得到，之后转换为整型值。

算法 3-1 单调对齐搜索 Monotonic Alignment Search

Data: 隐变量 z, 先验分布的统计变量 μ,σ, 梅尔频谱的长度 T_mel, 文本长度 T_text

Result: 单调对齐 A^*

初始化 $Q_{i,j} \leftarrow -\infty$：它表示模型看到先验分布统计量的前 i 维和隐变量 z 的前 j 帧时的对数似然函数。;

计算第一行数据 $Q_{i,j} \leftarrow \sum_{k=1}^{j} log\mathbb{N}(z_k;\mu_1,\sigma_1)$，对所有 j：;

while $j = 2$ ***to*** T_{mel} **do**

  while $i = 2$ ***to*** $min(j, T_{text})$ **do**

    $Q_{i,j} \leftarrow max(Q_{i-1,j-1}, Q_{i,j-1} + log\mathbb{N}(z_j;\mu_i,\sigma_i)$

  end

end

初始化 $A^*(T_{mel}) \leftarrow T_{text}$;

while $j = T_{mel} - 1$ ***to*** 1 **do**

  $A^*(j) \leftarrow argmax_{i \in A*(j+1)-1, A*(j+1)} Q_{i,j}$

end

3.1.4 对抗训练

引入判别器 D 判断输出是由解码器 G 输出，还是真实的波形。VITS 使用两种类型的损失形式，第一种是用于对抗训练的最小二乘损失函数（least-squares loss function）：

$$\begin{aligned} L_{adv}(D) &= \mathbb{E}_{(y,z)}[(D(y)-1)^2 + (D(G(z)))^2] \\ L_{adv}(G) &= \mathbb{E}_{z}[(D(G(z))-1)^2] \end{aligned} \tag{3-16}$$

第二种是特别施加于生成器的特征匹配损失（feature-matching loss）：

$$L_{fm}(G) = \mathbb{E}_{(y,z)}[\sum_{l=1}^{T}\frac{1}{N_l}||D^l(y) - D^l(G(z))||_1] \tag{3-17}$$

其中，T 表示判别器的层数，D^l 表示第层判别器的输出特征图（feature map），N_l 表示特征图的数量。特征匹配损失可以看作是重建损失，用于约束判别器中间层的输出。

3.2 总体损失函数

VITS 可以看作是 VAE 和 GAN 的联合训练，因此总体损失为：

$$L_{vae} = L_{recon} + L_{kl} + L_{dur} + L_{adv} + L_{fm}(G) \tag{3-18}$$

3.3 本章小结

本章首先介绍了 VITS 模型使用的数学原理，之后根据原理推理出了 VITS 模型训练的损失函数，在原理上解释了 VITS 的基本思想。

VITS 使用了 VAE，其能够对句子整体韵律特征实现有效地捕捉，而 Flow 能够重建音频细节特征，VITS 有效地整合使用两种算法，可以实现共享参数与优势；同时，VITS 有效地解决了传统 TTS 模型的 GAN 训练只针对声码器进行训练的问题，而 VITS 中的 GAN 是对各个模块都进行全局训练的。

第四章 对 VITS 模型测试设计方案

4.1 模型主要技术路线

VITS 的总体结构可以分为 5 块：

1. 先验编码器 (Textencoder)：文本编码器 + 提升先验分布复杂度的标准化流 f_θ。

2. 随机时长预测器 (StochasticDurationPredictor，SDP)：从条件输入 h_{text} 估算音素时长的分布。

3. 解码器 (Generator)：实际就是高保真生成对抗声码器[15]（Generative Adversarial networks for Efficient an High Fidelity Speech,Hifi-GAN) 的生成器。

4. 后验编码器 (PosteriorEncoder)：在训练时输入线性谱，输出隐变量 z，推断时隐变量 z 则由 f_θ 产生。VITS 的后验编码器采用 WaveGlow[14] 和 Glow-TTS[14] 中的非因果 WaveNet 残差模块。应用于多人模型时，将说话人嵌入向量添加进残差模块。仅用于训练。

5. 判别器 (Discriminator)：实际就是 HiFi-GAN 的多周期判别器。仅用于训练。

VITS 由于采用对抗训练的模式，模型主要包括生成器 *net_g* 和判别器 *net_d* 两大块，判别器仅在训练时使用。具体实现上，生成器 *net_g* 由 SynthesizerTrn 实现，包括先验编码器、随机时长预测器、解码器和后验编码器；判别器 *net_d* 由多周期判别器 (MultiPeriodDiscriminator) 实现，即 HiFiGAN 中的多周期判别器。

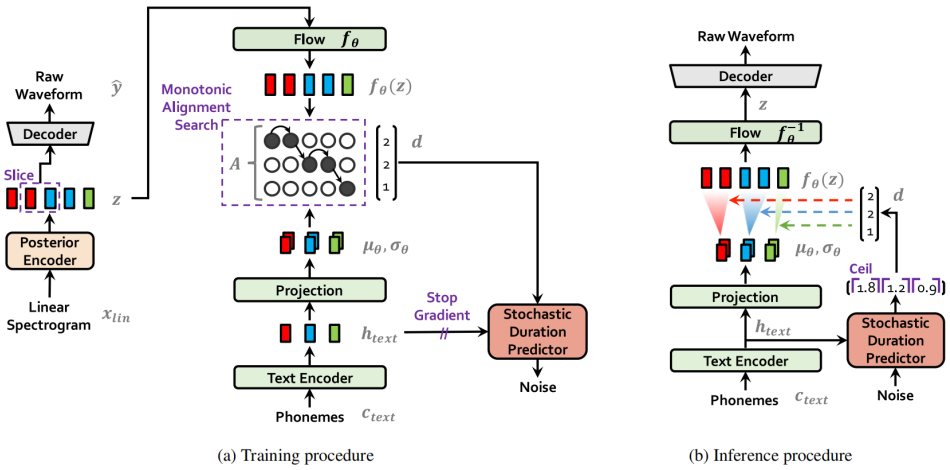


图 4-1 VITS 模型技术流程图

4.1.1 先验编码器

先验编码器包括文本编码器和标准化流。其中文本编码器由 Transformer blocks[16] 模型组成。其需返回 4 个值，用于计算音素时长的 x 和经过 Mask 矩阵遮挡操作的音素时长矩阵 *x_mask*，m 和 logs 经注意力权重加权求和之后，以残差形式求得先验隐变量。

Transformer Blocks 由多头注意力 (Multi-Attention,MA) 和前馈网络 (feedforward Neural Network,FNN) 组成：在 Transformer Blocks 的具体实现上，为了适应语音合成任务，利用相对注意力（Relative Attetion）鼓励自注意力关注临近的位置; 文本编码器的前馈网络由两层卷积组成，也就是卷积层 (conv1)->relu 层->dropout 层-> 卷积层 (conv2)，其中采用的卷积为等长卷积，输入为 n，输出也为 n，经过卷积层的特征抽取，输出特征数量不变，但是每个特征所包含的信息范围更广。

先验编码器中的标准化流，即残余耦合模块[17](ResidualCouplingBlock)，标准化流是 4 个仿射耦合层组成的模块，每个耦合层包含 4 个 WaveNet 残差结构，用于增强先验编码器的表达能力。WaveNet 的残差模块通过不断提高一维扩张卷积（带洞卷积）的膨胀系数，不断增大感受野，卷积后的结果一部分元素加到下一层的输入，另一部分元素加到最终的输出。

4.1.2 时长预测器

VITS 采用的随机时长预测器[1]（Stochastic Duration Predictor，SDP）输入的是文本编码器的结果 h_{text}, 而非文本编码器之后标准化流的输出隐变量 z，输出的是音素时长的对数。文本编码张量 h_{text} 首先通过前处理一维卷积，之后进入带洞深度可分离卷积[18]（Dialted and Depth-Separable Convolution，DDSConv），然后通过后处理一维卷积之后，最后进入神经样条流[19]（Neural spline flows）输出音素时长的对数。

DDSConv 在保持较大感受野的同时，提高参数利用效率，在 DDSConv 中，每一个卷积层之后都跟着层规范化和 Gelu 激活函数。具体来说，输入首先进入分组一维卷积，该分组卷积的组数和通道数相同，膨胀系数随着层数的递增而指数级增大，之后进入一维卷积层和 Dropout 层；多个分组卷积和一维卷积组成的模块构成了 DDSConv，每个模块的输出均作为残差元素加到输入上。

4.1.3 解码器

VITS 的解码器实际就是 HiFiGAN 的生成器，使说话人嵌入向量通过一维卷积元素加到隐变量 z 上主要是多组转置卷积，每组转置卷积后跟多感受野融

合[15]（Multi-Receptive Field Fusion，MRF）模块，所谓的多感受野融合模块主要是等大一维卷积组成的残差模块以提高合成语音时的推理速度。同时，HiFiGAN同时拥有多尺度判别器[15]（Multi-Scale Discriminator，MSD）和多周期判别器[15]（Multi-Period Discriminator，MPD），通过缩短序列长度，进行卷积，并将一维样本点序列以一定周期折叠为二维平面，进入多个卷积层，最终实现增强 GAN 判别器甄别合成或真实音频的能力。

4.1.4 后验编码器

VITS 的后验编码器包含 16 个 WaveNet 残差结构，输入线性对数幅度谱（linear-scale log magnitude spectrograms），输出 192 维的后验隐变量。在先验编码器的标准化流中同样使用了 16 层 WaveNet 残差结构。

4.1.5 判别器

VITS 采用对抗训练的方法，引入了 HiFiGAN 中的多周期判别器判别合成波形的质量。不同于 HiFiGAN 拥有多周期和多尺度判别器，为了提升训练效率，VITS 仅使用了多周期判别器。多周期判别器拥有多个子判别器，每个子判别器的重点是将一维样本点序列以一定周期折叠为二维平面，并在平面上进行二维卷积并输出样本为真实样本的概率

4.2 对 VITS 的研究内容

1. 对 VITS 语音合成模型的基本研究：掌握其模型构建方法，如何将 cVAE 和流算法应用于变分前验编码器得到隐变量；

2. 单调对齐搜索算法在基于生成流通过单调对齐算法的语音合成[14] 和 VITS 中的使用方法，了解其如何得到文本和梅尔频谱的最佳对齐；

3. 掌握 VITS 中如何使用变分反量化[20]（Variational Dequantization）和变分数据增广[21]（Variational Data Augmentation）来解决预测时的反量化和数据增广；

4. 在 VITS 的对抗训练中，了解如何用鉴别器 D[10]（discriminatorD）来分辨真实值和生成值的优势。

4.3 实施方案

1. 建立四川方言语音数据集；

2. 参考 VITS 模型，设计四川方言语音合成模型；

3. 使用 PyTorch(一个开源的机械学习库，用于自然语言处理) 实现并在单块

GPU 上训练四川方言语音合成模型。

4. 验证模型合成语音质量，设计消融实验 (Ablation Studies), 以平均意见得分为评估指标，验证标准流对先验编码器的编码效果的提升，以及后验编码器中使用线性频谱和梅尔谱的差异。

4.3.1 数据集制作

由于原始 VITS 论文中已经完成采用 LJspeech 数据集对英文 TTS 模型的构建，我所做的工作是将该模型迁移到四川话方言合成任务上。因此，我们需要制作一个四川方言数据集。我采用了李伯清评书语音作为语音集，使用了 audio-slicer 软件对音频进行了切割，并使用 Audition 批量转换格式（单声道、22050Hz，PCM 16bit，多数据块）以满足模型音频格式要求，最终得到了 330 条数据集语音和 42 条验证集语音，并制作了相对应的 flielist。

```
wavs/1_01.wav|今天讲点啥子？讲点我们四川人为啥子爱用叠词。什么叫叠词？
wavs/1_02.wav|有些人就弄不懂了，所谓叠词就是重叠的意思。比如说我们说：
wavs/1_03.wav|吃胰胰吃莽莽睡觉觉，这些都是叠词。
wavs/1_04.wav|甚至有些说哎叮叮猫，你两个钉子重起对不对。
wavs/1_05.wav|所以也叫第一次。那么话又说回来，说起这个叠词肯定有他一定的原因。
wavs/1_06.wav|第一个我在想，他首先一个用叠词的目的，
wavs/1_07.wav|是加强大家的记忆，对这个事情呢把它强调一下。
wavs/1_08.wav|二一个呢就显得轻松，显得活泼。当然话语说回来你如果说叮猫，
wavs/1_09.wav|也许你达到了你的目的，但是他就显得不是那么轻松。你比如说骂人。
wavs/1_10.wav|哈，瓜不兮兮的呀。瓜娃子娃娃不断的这样子骂。这个例子有些骂人，
wavs/1_11.wav|他骂得使你接受不了，对不对如果你显得很生硬，瓜娃子你还硬起。
wavs/1_12.wav|瓜兮兮。他有点儿朝底下垮，他不是硬起。
wavs/1_13.wav|瓜娃子。他就硬起的。但他说瓜兮兮，他就不可能一下飘那么高。
wavs/1_14.wav|他自然就往下面落，人就显得轻松一些。
```

图 4-2 自制 flielist 示意图

同时，由于是自制数据集，数量不够多，且数据质量层次不齐，有许多的杂音，如果直接进行训练，四川话的音素与文本对齐难度较大，容易产生梯度消失的问题，导致训练出来的模型难以合成出高质量的四川方言。为了解决该训练问题，我将数据集与标贝中文女声语音数据集进行了联合训练，以解决梯度消失的问题。我使用了标贝中文女声语音数据集进行模型预训练。该数据集包含了约 10000 条长度约 5 秒左右的无噪声语音，情感波动较少，非常适合于用于生成中文语音合成模型前的预训练。标贝还提供了相应的文本文件进行训练。我进行了 275 轮的训练，取得了较为优秀的结果。语音纯净度高，拟真度也较高。以此为基准，换用准备好的四川话数据集进行训练，训练了 2400 轮，得到了比较理想的合成效果。

4.3.2 中文预处理文本

由于 VITS 论文中使用的是英语模型，而我们所做的是一种中文模型，因此需要对原始的中文汉字符号进行整理。数据中除了中文字符外，还包含数字、英文字符、标点等无意义的非常规字符，需要进行处理。我使用了正则表达式对中文文本进行预处理和清理。分词是中文文本分析中的关键步骤，正确的分词能够更好地构建模型并应用算法进行分析。在此，我使用了五笔输入法符号表示原始的中文词语，并将这些符号映射到计算机可识别的 ASCII 码。这样做不仅提高了模型的性能，还能帮助实现更高的准确性。 通过中文预处理模块，我们将中文字符

图 4-3 中文预处理模块示意图

全部转化为中文五笔符号，既能够实现归一化的操作，且相比于直接用拼音，能够将不同含义的同音词区分开，像“建模”和“剑魔”这样的词语，如果用拼音则无法很好的进行表征，会导致训练时语调出现错误，而使用五笔符号能够更好的解决一对多的问题。

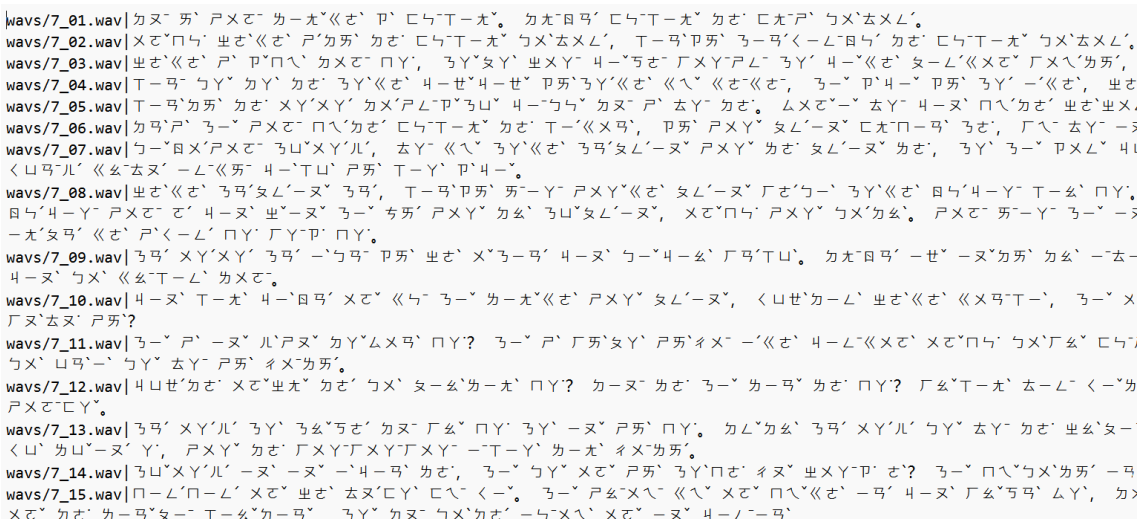


图 4-4 预处理后的 flielist 示意图

4.3.3 训练环境配置

2006 年，NVIDIA 公司发布了统一计算设备架构 (Compute Unified Device Architecture，CUDA)，是一种新的通用并行计算架构。它实现了在 GPU 上进行并行计算和编程，利用 GPU 的并行计算引擎来更加高效地解决比较复杂的计算难题。解决的是用更加廉价的设备资源，实现更高效的并行计算。CUDA 的深度神经网络库 (cuDNN) 是一个 GPU 加速的深度神经网络基元库，能够以高度优化的方式实现标准例程（如前向和反向卷积、池化层、归一化和激活层）。

PyTorch 是一种用于构建深度学习模型的功能完备框架，是一种通常用于图像识别和语言处理等应用程序的机器学习。该框架将 Torch 中高效而灵活的 GPU 加速后端库与直观的 Python 前端相结合，后者专注于快速原型设计、可读代码，并支持尽可能广泛的深度学习模型。其独特之处在于，它完全支持 GPU，并且使用反向模式自动微分技术，因此可以动态修改计算图形。这使其成为快速实验和原型设计的常用选择。

由于自己的电脑（GPU：GTX1060 6G，CPU：R5 5600，内存 16G）配置不够，如果强行用于环境配置进行训练，批量大小只能设置为 4 以下才能满足显存容量需求，这对训练效率是非常不利的。因此我放弃了本地部署，选择使用了虚拟镜像环境，在 autoDL（一个国内的算力市场）上租用了更好的训练环境（RTX3090 24G，CPU Intel(R) Xeon(R) Silver 4210R CPU @ 2.40GHz，内存 56G），可将批量大小设置为 16 以上，以满足高效的模型训练。

由于 VITS 源码使用的是 pytorch 环境，因此我们需要安装 conda 虚拟环境以及 Torch，以及更新 CUDA 版本，并安装相应依赖。我们选用的镜像文件是 PyTorch 2.0 + Python 3.8 + Cuda 11.8。

在 VITS 训练阶段，采用了 AdamW[22] 优化器，在训练过程中设置了权重衰减与 L2 正则化的方式，以减少“过山车”式的梯度下降，能够避免或减少收敛速度过慢的问题；同时设置了保存间隔（eval_interval），按照默认的 1000 进行模型的保存即可以满足保存的需求，设置过小会训练过程会耗费大量时间在保存上；设置过大如果训练出现问题无法满足及时保存最近的模型的需求；batch_size 设置为 32 以便充分使用显卡能力；同时，针对训练过程，cuDNN 需要启用以提高训练速度；使用半精度浮点数（float16，fp16）训练可以解决或者缓解上面 fp32 对显卡要求较高的问题：fp16 仅有 16bit，2 个字节组成，显存占用更少，通用的模型 fp16 占用的内存只需原来的一半，训练的时候可以使用更大的 batch_size, 获得更快的训练速度；同时由于 fp16 也会带来溢出错误（Grad Overflow）和舍入误差（Rounding Error）的问题，混合精度训练[23](Mixed Precision training) 方法被使用，

通过在内存中用 FP16 做储存和乘法从而加速计算，用 FP32 做累加避免舍入误差的方式，有效地缓解了舍入误差的问题；同时通过损失放大[23](Mixed Precision training)（loss scaling）方法，在反向传播过程中将损失变化（dLoss）手动增大 2^k 倍，以减少无法收敛的问题。在 Pytorch 中，我们调用 NVIDIA 设计的 API：自动混合训练（Automatic Mixed Precision，AMP）来实现这一功能。

由于这个语音成是一个大规模的深度学习模型，

4.3.4 对比实验设计

在实验组得设置上，我们先用经历过中文女声数据集预训练过的模型进行对四川方言的训练为了证明 VITS 合成语音的高表现力，我设计了对比实验。我采用了 tacotron2+HiFiGAN 语音合成模型所合成出的语音，以及真实语音进行对比实验。其中，我是用了谷歌 Colab 平台进行训练部署，对 tacotron2+HiFiGAN 语音合成模型进行了 360 轮训练，得到了相对较好的效果。

同时，我们还设计了消融实验（ablation study），消融实验类似于控制变量方法，常用于神将网络中用于验证某一模块的效果。我们用其来验证预训练模型对于最终语音合成质量的提升。我们设计了另一组对照组，该组直接进行对四川方言的训练而不使用预训练模型，在相同环境下使用 VITS 模型进行相同数量的训练轮次（360 轮），最终与经过预训练的模型进行对比 MOS 值。

本实验通过主观方法和客观方法来验证模型性能，在主观方法中，我们我们设计了调查问卷，使用每个模型生成的 10 条语音，让 20 名中文母语者对听到的语音进行打分，最后计算出各个模型的平均意见得分（MOS）。这 20 名测试人员并不全是掌握四川方言的人，且来自中国各个省份，因此具有比较强的随机性，调查结果更具说服力。

4.3.5 tacotron2+HiFiGAN 语音合成模型

Tacotron2 是由谷歌在 2017 年提出来的一个端到端语音合成框架。其模型主要由三部分组成：

声谱预测网络：一个引入注意力机制[16]（Attention）的基于循环的语句对语句（Sequence to Sequence，Seq2seq）的特征预测网络，用于从输入的字符序列预测梅尔频谱的帧序列。

声码器 (vocoder): 在原文中使用了 Wavenet 的修改版本来预测的梅尔频谱帧序列来生成时域波形样本。如今，更优秀的 HiFiGAN 已经逐渐取代 Wavenet 来提供声码器的功能。

中间连接层：使用低层次的声学表征-梅尔频率声谱图来衔接系统的两个部分。

其模型如下图表示：

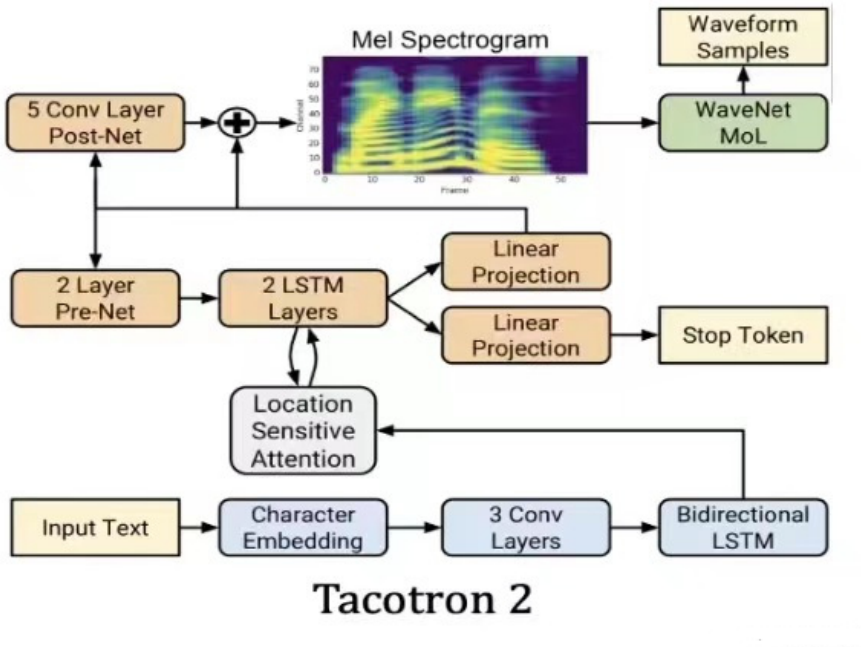


图 4-5 Block diagram of the Tacotron2 system architecture

网络由一个编码器（蓝色）和一个解码器（橙色）组成。编码器将一个字符序列转换成一个隐藏的特征表示，作为解码器的输入来预测 spe CTR 图。输入文本（黄色）是使用学习的 512 维字符嵌入来呈现的，它通过三个卷积层（每个包含 512 个形状为 5 × 1 的滤波器）的堆栈，然后进行批量规范化和 ReLU 激活。编码器输出被传递到注意力网络（灰色），该网络将完整编码序列总结为每个解码器输出步骤的固定长度上下文向量。

解码器是一个自回归递归神经网络，它从编码的输入序列中一次一帧地预测 mel-spe CTR 图。前一个时间步的预测首先通过一个包含两个完全连接的 256 个隐藏 ReLU 单元的层的小 pre 网络。prenet 输出和注意力上下文向量被连接起来，并传递到一个由两个 LSTM 层组成的堆栈，其中包含 1024 个单元。通过线性变换，将 LSTM 输出与注意上下文向量的连接进行投影，以预测目标 spe CTR 图帧。最后，将预测的 mel-spe CTR 图通过一个 5 层卷积后网络，该网络预测一个残差来加入预测，以改善整体重建。每个 post-net 层由 512 个形状为 5 × 1 的过滤器组成，并进行批量标准化处理，除最后一层外，所有过滤器均激活。

Tacotron2 中使用了基于位置敏感的注意力机制 [24]，利用注意力权重来降低解码过程中的错误。对位置特征的计算：位置特征用 32 个长度为 31 的 1 维卷积核卷积得出，由之前的累加的注意力权重经过卷积得来，然后把输入序列和为位置

特征投影到隐层表征，计算出注意力权重。

Tacotron2 选择预测梅尔频谱，使用的是混合注意力机制，在对齐中加入了位置特征。

需要注意的是，Tacotron2 并不是直接生成出音频的端到端系统，而是生成出梅尔频谱，再通过其他声码器进行声学特征的提取，合成出人耳能够听到的音频。在本文中，我们采用 HiFiGAN 作为声码器进行合成，HiFiGAN 是近年来在学术界和工业界都较为常用的声码器，能够将声学模型产生的频谱转换为高质量的音频，相比于使用 World 声码器 [25] 和 Wavenet [4] 声码器，其拥有更好的性能。

由于 Tacotron 不支持直接的汉字输入，因此我们还需要把汉字转换为拼音才能够供该模型识别。因此对数据集要进行一个转换：

```
Paimon/testing/7_01.npy|dou1 ai4 shuo1 liang3 ge4 zi4 fen1 xiang3  dang1 ran2 fen1 xiang3 de fang1 shi4
bu4 tong2
Paimon/testing/7_02.npy|wo3 men zhe4 ge4 shi2 dai4 de fen1 xiang3 bu4 tong2  xian4 zai4 nian2 qing1 ren2
de fen1 xiang3 bu4 tong2  bi3 ru2 wo3 men zhe4 ge4 shi2 de fen1 xiang3
Paimon/testing/7_03.npy|zhe4 ge4 shi4 zi3 mei4 duo1 ma  na3 pa4 zhua1 ji3 ke1 hua1 sheng1 na2 ji3 ge4
ping2 guo3 hui2 lai2  shou3 xian1 jiu4 yao4 gei3
Paimon/testing/7_04.npy|xian1 ba3 da4 de na4 ge4 jie3 jie3 zai4 na2 ge4 gei3 ge1 ge1  ni3 zi4 ji3 zai4
na2 yi2 ge4  zhe4 ge4 ke3 yi3 da4 jia1 fen1 xiang3
Paimon/testing/7_05.npy|xian4 dai4 de wa2 wa2 du2 sheng1 zi3 nv3 ji1 ben3 dou1 shi4 ta1 de  suo3 yi3 ta1
```

图 4-6 汉字转拼音模块

4.3.6 对比实验环境配置

与 VITS 相同，本地训练环境配置较差，因此我们选用谷歌 Colaboratory 进行对 Tacotron2 模型的训练。Colaboratory 是一个 Google 研究项目，旨在帮助传播机器学习培训和研究成果。它是一个 Jupyter 笔记本环境，不需要进行任何设置就可以使用，并且完全在云端运行。Colaboratory 笔记本存储在 Google 云端硬盘中，并且可以共享，就如同使用 Google 文档或表格一样。Colaboratory 可免费使用。利用 Colaboratory ，可以方便的使用 Keras,TensorFlow,PyTorch,OpenCV 等框架进行深度学习应用的开发。在本次实验中，我采用了免费版本的 T4 显卡。

在超参数的设置上，为了防止爆显存等问题，我们仅选用 16 的 *batch_size* 进行训练，训练了 360 轮。在各种参数的设置中，注意力机制的 *dropout_rate* 设置的值为 0.1，decoder 的 dropout_rate 设置的值为 0.1；设置的最大学习率为 0.0005，最小学习率为 0.00001，在训练过程中逐渐减小学习率；同样也采用 fp16 半精度训练，以对模型训练进行加速。最终经过 360 轮的训练以及 7417 次的迭代后 Validation loss 达到了 0.458366 得到了比较好的合成效果。下图展示了训练到 7417 次迭代之后的注意力对齐效果：

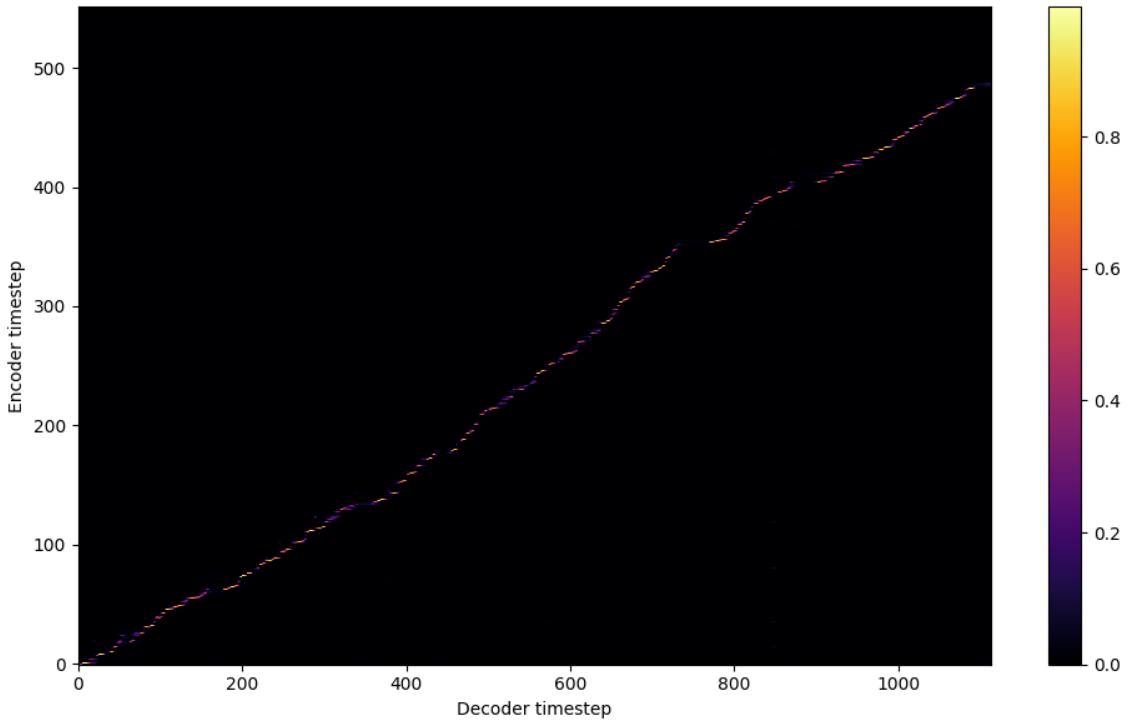


图 4-7 Attention alignment of Tacotron2

4.4 出现问题与解决方法

1）报错“RuntimeError: Expected to have finished reduction in the prior iteration before starting a new one. This error indicates that your module has parameters that were not used in producing loss.”，这是由于 Pytorch 自带的 API：分布式数据训练（DistributedDataParallel，DDP）中有一个参数 *find_unused_parameters = False*, 如果某个参数的梯度没有 n 份（n 为分布式训练的节点总数），这个参数的梯度将不会被平均（每个节点的梯度都不一样，将导致各个节点的参数发散），这会导致代码报错，这是由于 pytorch 的 DDP 训练模式的隐藏要求是每个节点的梯度必须相同，这样才能保证每个节点的模型参数相同。而如果我们将其改为 True，则可以找到每一个分布式节点没有梯度的参数，然后进行平均，以保证每一个节点梯度相同。

2）ZeroDivisionError: integer division or modulo by zero 的问题这是因为，VITS 源码中调用了分桶操作（DistributedBucketSampler），作用是将数据集的各种长度的音频放入不同的“桶”中进行分布式 DDP 训练，如果某一个音频长度没有存在于任一个桶的长度区间中，就会导致这个桶的梯度为 0。为了解决该问题，我对数据集和验证集数量进行了增加，并降低了 batch_size 的大小，同时增加了分桶区间，这样可以保证音频采集函数采集的数量不小于 batch_size 的设置值，从而避免

得到的张量长度不足，进而保证分桶分布式训练操作正常执行。

3）在对四川话预料进行训练过程中，我遇到了 WavFileWarning: Chunk (non-data) not understood, skipping it. 的警告。主要原因是 WAV 格式不对，需要使用旧版 WAV 编码而不是新版 RF64 编码。通过 FFMPEG 进行编码更改后，成功消除了改警告。

4.5 本章小结

在本章中，本文详细的阐明了 VITS 的总体技术路线，对各个结构的神经网络模块以及采用算法和设计进行了详细的说明；接着介绍了本文的研究内容，掌握了 VITS 的结构特点，在结构上展现了 VITS 模型设计的成功之处，利用 VITS 合成出四川方言语音；最后本文介绍了使用对比实验和消融实验对 VITS 模型性能进行评估，将 VITS 模型生成语音与 Tacotron2 和未进行预训练的模 VITS 模型进行主观和客观的性能对比，以验证该模型的优秀之处。

第五章 实验结果

5.1 语音合成评价指标

对合成语音的质量评价，主要可以分为主观和客观评价。客观评价是通过一些数字信号处理模块，模拟出人类的听觉，展示频谱细节，在语音转换技术中常常通过计算梅尔倒谱失真[26]（Mel Cepstral Distortion，MCD）等方法作为客观评价。主观评价是通过人类主观的听感对语音质量进行打分，其中最常用的评价指标是平均意见得分[27]（Mean Opinion Score，MOS），这也是本文所采用的评价标准。

5.2 客观评价

语音转换的客观评测标准是梅尔倒谱失真，它表示的是转换后语音的梅尔倒频谱系数[26]（Mel-scale FrequencyCepstral Coefficients，MFCC）特征与标准输出语音的 MFCC 特征的差距。MCD 越低，失真越小，意味着两个音频段之间的相似度越高。

5.2.1 梅尔倒频谱系数

MFCC 依据人的听觉实验结果来分析语音的频谱。我们知道，频率的单位是赫兹（Hz），人耳能听到的频率范围是 20-20000Hz，但人耳对 Hz 这种标度单位并不是线性感知关系。其关系可以由这个式子表示：

$$mel(f) = 2595 * log_{10}(1 + f/700) \tag{5-1}$$

下图为 Mel 频率与线性频率的关系：

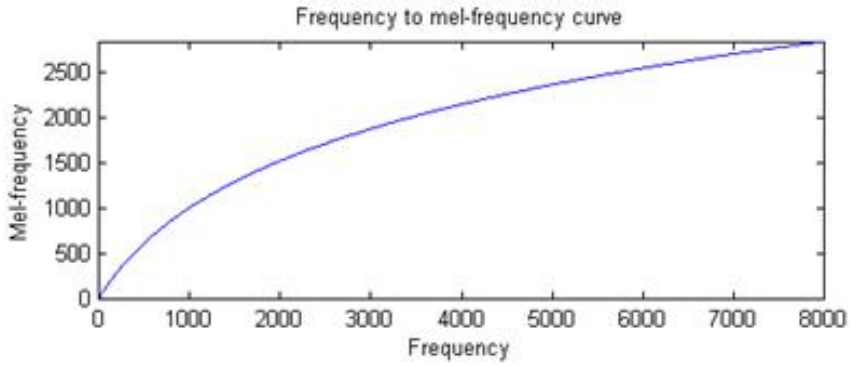


图 5-1 The relationship between Mel frequency and real frequency

其表明，在 Mel 频域内，人对音调的感知度为线性关系。举例来说，如果两

段语音的 Mel 频率相差两倍，则人耳听起来两者的音调也相差两倍。根据人耳听觉机理的研究发现，人耳对 200Hz 到 5000Hz 的频率敏感程度要大于更高频的声音。同时，人耳在收集声音过程中也会经历遮掩效应的影响，频率较低的声音容易遮掩频率较高的声音。这说明了人耳对低频音调的感知较灵敏，在高频时人耳是很迟钝的。所以，为了模拟出人耳感知声音的机制，人们设计了一组由低频到高频频带内按临界带宽的大小由密到疏的带通滤波器，对输入声音信号进行滤波处理。声音经过每个带通滤波器输出后的能量可以作为这个频段声音的基本特征，对此特征经过进一步处理后就可以作为语音的输入特征。而在 MFCC 中，常常通过傅里叶变换及逆变换提取出倒谱来获得原始音频的低频包络细节特点。

由于 MFCC 对输入信号的性质没有任何要求，同时，基于人耳的听觉模型非常适用语音识别及语音合成领域。因此，这种参数比基于声道模型的线性预测倒谱系数[28](linear predictive cepstral coefficient, LPCC) 相比，鲁棒性显著提高，更符合人耳的听觉特性，而且当信噪比降低时，仍然具有较好的识别性能。因此逐步成为大多数语音处理领域的必选研究指标。

MFCC 提取过程包括：

1）先对语音进行预加重、分帧和加窗；预加重处理其实是将语音信号通过一个高通滤波器；

$$H(z) = 1 - \mu z \tag{5-2}$$

式中 μ 的值介于 0.9-1.0 之间，我们通常取 0.97。预加重的目的是提升高频部分，使信号的频谱变得平坦，保持在低频到高频的整个频带中，能用同样的信噪比求频谱。同时，发声过程中会产生声带和嘴唇的效应，因此我们需要对该部分的能量进行消除，并补偿语音信号受到发音系统所抑制的高频部分，也为了突出高频的共振峰。

2）将信号分帧后，我们需要通过加窗来消除各个帧两端造成的信号频谱泄漏（spectral leakage）问题，我们需要对每一个短时分析窗，通过快速傅里叶变换（Fast Frontier Transfrom,FFT）得到对应的频谱。常用的窗函数有方窗、汉明窗和汉宁窗等，根据窗函数的频域特性，常采用汉明窗（hamming window）。汉明窗函数:

$$W(n,a) = (1-a) - a \times cos[\frac{2\pi n}{N-1}] \tag{5-3}$$

其中 a=0.46,N 为每一帧的长度。加窗操作：

$$S'(n) = S(n) \times W(n) \tag{5-4}$$

3）将上面的频谱通过 Mel 滤波器组得到 Mel 频谱。mel 滤波器的实现过程：首先，确定最低频率和最高频率，以及梅尔滤波器个数；通过公式 5-1 转换最低频率和最高频率的相对梅尔频率，然后计算相连两个 Mel 滤波器中心 Mel 滤波器的距离；在 Mel 滤波器上，两两之间的中心频率等间距的：

$$d_{mel} = \frac{high_{mel} - low_{mel}}{M+1} \tag{5-5}$$

然后将这些等间距的中心 Mel 频率转化为非等间距的频率 f 上；最后计算频率所对应的 FFT 中点的下标：

$$bin = \frac{hz_{point}}{fs/2} \cdot (\frac{fft_{len}}{2} + 1 = \frac{(fft_{len}+2) \cdot hz_{point}}{fs} \tag{5-6}$$

4）对数运算：计算每个滤波器输出的对数能量：

$$s(m) = ln(\sum_{k=0}^{N-1} |X_a(k)|^2 H_m(k)), 0 \le m \le M \tag{5-7}$$

其中 M 为设计的滤波器个数。取绝对值是仅使用幅度值，忽略相位的影响。对数运算是为了分别包络（envelope）和细节（details），包络代表音色，细节代表音高，因此当我们需要指标来描述音频的音色特点时，我们可以通过对数运算提取包络实现，MFCC 也是如此。另外，人的感知与频率的对数成正比，正好使用对数谱进行模拟。

5）经过离散余弦变换（Discrete cosine transform，DCT）得到 MFCC 系数。D 较 DFT 变换具有更好的频域能量聚集度，对于不重要的频域区域和系数能够直接舍弃掉，因此，DCT 变换非常适合于图像压缩算法的处理和音频信号处理音频信号处理领域。DCT 方法如下：

$$C(n) = \sum_{m=0}^{N-1} s(m) cos(\frac{\pi n(m-0.5)}{M}), n = 1, 2, ..., L \tag{5-8}$$

将上述的对数能量带入离散余弦变换，求出 L 阶的 Mel-scale Cepstrum 参数。这里 M 是三角滤波器个数。6) 动态特征提取标准的倒谱参数 MFCC 只反映了语音参数的静态特性，语音的动态特性可以用这些静态特征的差分谱来描述。实验证明：把动、静态特征结合起来才能有效提高系统的识别性能。差分参数的计算可以采用

下面的公式：

$$d_t = \begin{cases} C_{t+1} - C_t & ,t < K \\ \frac{\sum_{k=1}^{K} k(c_{t+k} - C_{t-k})}{\sqrt{2\sum_{k=1}^{K} k^2}} & ,others \\ C_t - C_{t-1} & ,t \geq Q - K \end{cases} \tag{5-9}$$

式中,dt 表示第 t 个一阶差分，Ct 表示第 t 个倒谱系数，Q 表示倒谱系数的阶数，K 表示一阶导数的时间差，可取 1 或 2。将上式的结果再代入就可以得到二阶差分的参数。

因此，MFCC 的全部组成其实是由：N 维 MFCC 参数（N/3MFCC 系数 + N/3 一阶差分参数 + N/3 二阶差分参数）+ 帧能量组成。

6）得到 MFCC 之后，只需要对原始语音和合成语音进行帧对齐处理，便可以计算出合成语音与真实语音之间的相似程度。

5.2.2 梅尔倒谱失真

设某一帧的标准输出特征为 y，而转换后语音的特征为 $\hat{y}$，则这一帧的 MCD 定义为：

$$\mathrm{MCD}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{10\sqrt{2}}{\ln 10} ||\boldsymbol{y} - \hat{\boldsymbol{y}}||_2 \qquad (1) \tag{5-10}$$

MCD 的单位为分贝（dB）。上式前面的系数，就是为了把单位转化成分贝的，其中除以是为了把本身就是自然对数值的 MFCC 转换成常用对数。MCD 的优点是可以全自动计算，但研究发现，它与人们主观感受到的音质的相关性并不够强。因此，语音转换更可靠的评测方法是主观评测。

5.3 主观评价

主观评价中的 MOS 评测基于人类的主观听觉感受，可以衡量语音的自然度，流畅度，及与真实语音的相似度。其中绝对等级评分（Absolute Category Rating，ACR）应用最为广泛，ACR 的详细评估标准如下表所示。

表 5-1 主观意见得分的评估标准

优	5.0	很好，听得清楚；延迟小，交流流畅
良	4.0	稍差，听得清楚；延迟小，交流欠流畅，有点杂音
中	3.0	还可以，听不太清；有一定延迟，可以交流
差	2.0	勉强，听不太清；延迟较大，交流需要重复多遍
劣	1.0	极差，听不懂；延迟大，交流不通畅

在使用 ACR 方法对语音质量进行评价时，参与评测的人员（简称被试）对语音整体质量进行打分，分值范围为 1 到 5 分，分数越大表示语音质量越好。MOS 大于 4 时，可以认为该音质是被大众认可的，在通信及日常交流中能够被识别；若 MOS 低于 3，则说明该语音存在比较大的问题，大部分被试并人员不满意该音质。

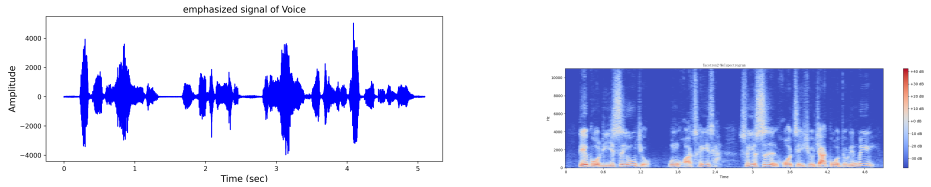
5.4 声谱图

声音信号本是一维的时域信号，直观上很难看出频率变化规律。如果通过傅里叶变换把它变到频域上，虽然可以看出信号的频率分布，但是丢失了时域信息，无法看出频率分布随时间的变化。为了解决这个问题，很多时频分析手段应运而生，短时傅里叶变换 (Short Time Fourier Transform，STFT）是最经典的时频域分析方法。STFT 通过把一段长信号分帧、加窗，再对每一帧做傅里叶变换，最后把每一帧的结果沿另一个维度堆叠起来，得到类似于一幅图的二维信号形式。如果我们原始信号是声音信号，那么通过 STFT 展开得到的二维信号就是所谓的声谱图（spectrogram）。一段语音被分为很多帧，每帧语音都对应于一个频谱（通过短时 FFT 计算），频谱表示频率与能量的关系 (不同频率的振幅大小不同)。振幅用颜色深浅来表示，振幅越大，颜色越深，即把幅度映射到一个灰度级表示 (0 表示白，255 表示黑)，幅度值越大，相应的区域越黑。

声谱图（Spectogram）是声音频率随时间变化的频谱的可视化表示，是给定音频信号的频率随时间变化的表示。STFT 将数据转换为时频信号。通过 STFT 转换信号，以便我们可以知道给定时间给定频率的幅度。使用 STFT，我们可以确定音频信号在给定时间播放的各种频率的幅度。

对于 Tacotron2+HifiGAN 模型和 VITS 模型，我使用同一语句，对两种模型生成的合成语音频谱图进行比较，生成相对应的频谱图，比较两者是否完美复原语句中的各个音素。语句为“德国历史悠久、文化璀璨，素有“诗人和哲学家国度”的美誉”。

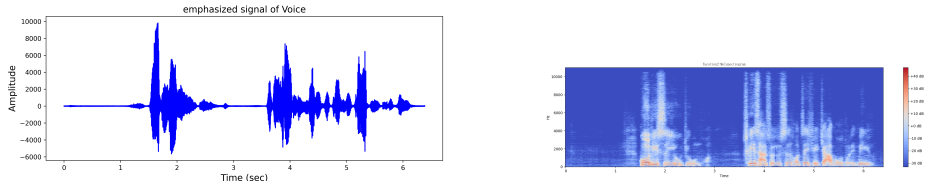
VITS 模型合成语音结果如下：



(a) (b)

图 5-2 VITS-spectrogram

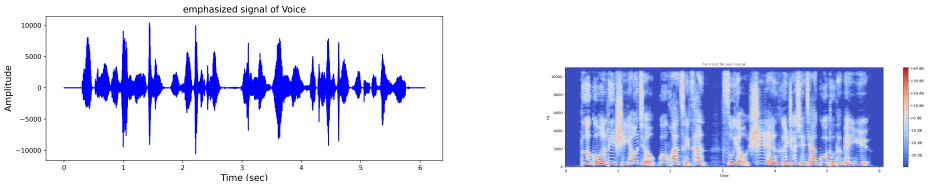
Tacotron2+HIFIGAN 模型合成语音结果如下：



(a) (b)

图 5-3 Tacotron2+HIFIGAN-spectrogram

真实语音如下：



(a) (b)

图 5-4 Ground Truth Wave-spectrogram

可以发现，VITS 模型合成语音整体波形比较平稳，对整体时间长度和各个音素的的时间长度预测比较准确。而 tacotron2+HIFIGAN 模型明显有音素信息缺失，且存在重复吐词或漏词现象。这是由于 Tacotron2 模型是自回归模型，自回归模型的文本输入端与梅尔频谱输出端之间的联系主要由注意力机制 [16] 去保证，而注意力机制的本质在于使得机器自身去学习文本与语音之间的映射关系。但注意力机制的映射存在不稳定性，导致该情况的发生。而 VITS 采用单调对齐搜索算法进行文本域音频之间的对齐，同时利用随机时长预测器通过梯度下降的方法来逐渐训练音素市时长。此外，VITS 在语速上更接近原始音频，也证明了 VITS 模型的强大预测能力。

5.5 MFCC 实验设计

我们设计了一个拥有 24 个滤波器的非线性分布的梅尔滤波器组，采用滤波器为三角滤波器，在“梅尔刻度”上，及公式 5-1 中的梅尔频率与真实频率的映射关系上，这 24 个三角带通滤波器的频率上是平均分布的，及人耳对于频率 f 的感受是呈对数变化的。其结构可如下图所示：

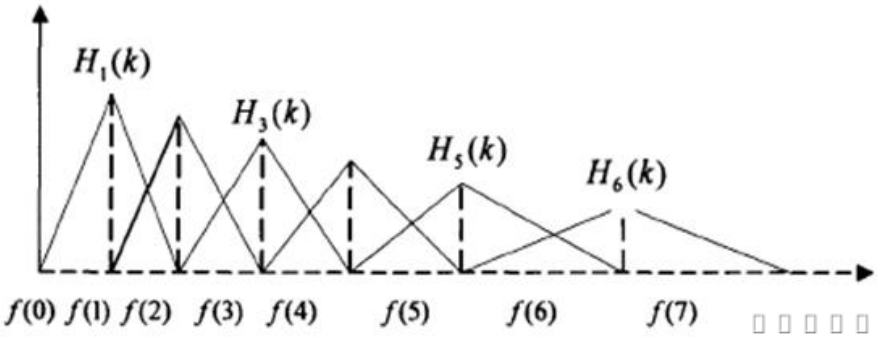


图 5-5 The illustration of Mel-filter

这可以通过一下等式建模，三角滤波器的频率相应定义为：

$$f_m(k)=\begin{cases}0 & ,k<f(m-1)\\ \dfrac{2(k-f(m-1))}{(f(m+1)-f(m-1))(f(m)-f(m-1))} & ,f(m-1)\le k\le f(m)\\ \dfrac{2(f(m+1)-k)}{(f(m+1)-f(m-1))(f(m)-f(m-1))} & ,f(m)\le k\le f(m+1)\\ 0 & ,k\ge f(m+1)\end{cases} \tag{5-11}$$

三角带通滤波器有两个主要目的：1）三角形是低频密、高频疏的，这可以模仿人耳在低频处分辨率高的特性；

2）频谱有包络和精细结构，分别对应音色与音高。在每个三角形内积分，就可以消除精细结构，只保留音色的信息。（因此一段语音的音调或音高，是不会呈现在 MFCC 参数内，换句话说，以 MFCC 为特征的语音辨识系统，并不会受到输入语音的音调不同而有所影响）通过三角形滤波器组对频谱进行平滑化，并消除谐波的作用，突显原先语音的共振峰。3）傅里叶变换得到的序列很长（一般为几百到几千个点），把它变换成每个三角滤波器下的能量，可以减少运算量。我们设置了一个 24 组得 mel 滤波器，FFT 个数是 256，采样率为 8000，滤波器频率范围的最低频率$f_l=0$, 滤波器频率范围的最高频率$f_h=f_s/2=8000/2=4000$，根据公式 5-1 将频率转换到 mel 频率，获得的等幅滤波器组如下：

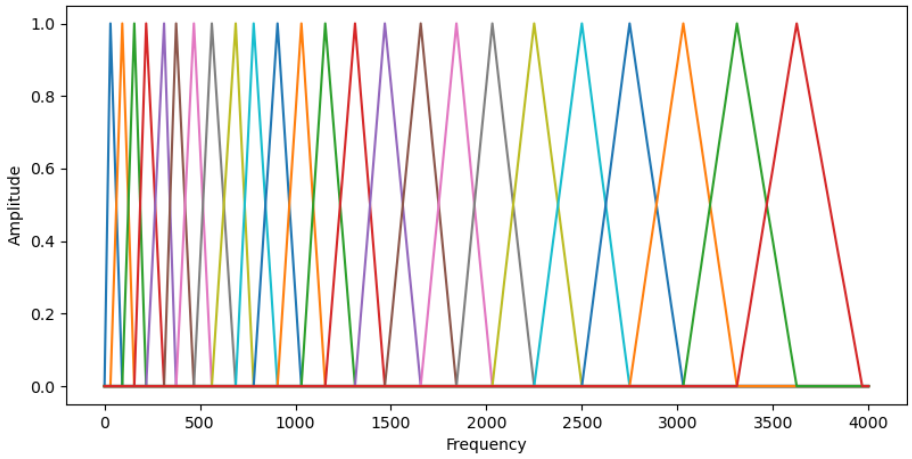
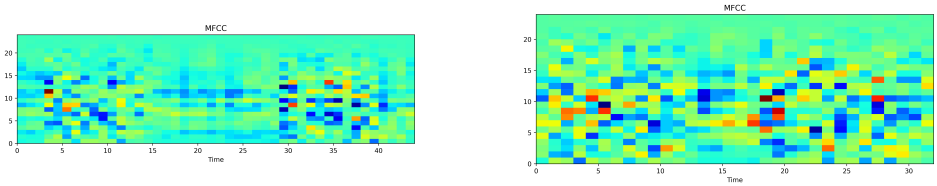


图 5-6 Filter bank on a Mel-Scale

我们的测试语句是“都爱说两个字分享，当然分享的方式不同”。经过后续离散余弦变换 (Discrete Cosine Transform,DCT) 变换，我们得到了，VITS 合成语音和 Tacotron2+HiFiGAN 合成语音的 MFCC。结果如下：



(a) (b)

图 5-7 MFCCs of Tacotron2+HiFiGAN(左) & VITS(右)

然后将原始音频经过同样方法生成出其 MFCC:

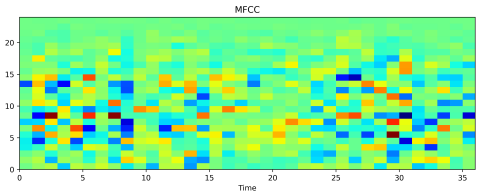


图 5-8 MFCCs of Ground-Truth Speech

可以明显的观察到，VITS 模型的 MFCC 在每一帧上与原始模型更加近似，而 Tacotron2 在中间部分（16 帧到 30 帧）有明显的音素信息丢失。因此可以说明，VITS 模型的语音合成结果更优于 Tacotron2+HiFiGAN 模型。

5.6 MCD 实验设计

我们将上述的模型生成的 MFCC 和原始音频的 MFCC 带入公式（5-1）进行计算，求出相对应的 MCD 系数；由于生成语音长度是由模型预测的，不完全等同于原始音频长度，为了得到相同的系数矩阵，我仅提取各个音频的前 3s 进行计算。由于采用的 MFCC 维度是 24 维，则最终各个音频的 MFCC 系数都为 30x24 维的矩阵。得到的最终结果如下：

表 5-2 各模型的 MCD

Model	MCD(dB)
Tacotron2 + HiFi-GAN	55.67
VITS	28.40

评估结果显示，VITS 更接近原始音频，客观的说明了 VITS 的高表现力。

5.7 平均主观得分

本文采用了众包的 MOS 测试来评估合成语音质量。20 位评审员听取由各个模型随机生成的 10 条合成语句以及真实语句，且这些生成的语音不来自语音训练集，而是随机产生的语句。我们对所有的音频片段进行了标准化处理，以避免振幅差异对评分的影响。结果如下：

表 5-3 在 95% 置信程度下，使用四川方言数据集，各模型的 MOS 得分

Model	MOS（CI）
Ground Truth	4.77(±0.06)
Tacotron2 + HiFi-GAN	3.24(±0.08)
VITS	4.11(±0.06)
VITS(without pre-training)	3.44(±0.06)

评估结果显示，VITS 的表现优于使用 Tacotron2+HiFiGAN 的 TTS 系统，并取得了与 Ground-truth 语音相似的 MOS 值。此外，经过预训练之后的 VITS 模型生成的语音质量明显优于未预训练过的 VITS 模型。这些结果意味着：1）随机时长预测器产生比确定性时长预测器更真实的音素时长；2）我们的端到端训练方法是一种有效的方法，可以比其他 TTS 模型产生更好的样本。3）我们的端到端训练方

法是一种有效的方法，即使保持类似的时长预测结构，也能比其他 TTS 模型做出更好的预测器结构。通过设计预训练模型可以非常有效地避免过拟合和梯度消失等问题，通过快速的找到中文发音的收敛位置，大大提升了训练效率。

此外，为了验证 VITS 在合成四川方言中效果不逊色于合成其他语言，我们使用了标贝中文普通话数据集和 LJspeech 英文数据集作为参照，进行了比较实验。同样的，我们请 20 位评审员听各个模型生成的 10 条不同语言的合成语音，生成的语句不来自训练数据集，而是随机生成。结果如下：

表 5-4 在 95% 置信程度下，使用不同数据集，各模型的 MOS 得分

Model	Dataset	MOS（CI）
Tacotron2 + HiFi-GAN	LJspeech	3.82(±0.08)
Tacotron2 + HiFi-GAN	标贝中文	3.88 ± 0.08)
Tacotron2 + HiFi-GAN	四川方言	3.24(±0.08)
VITS	LJspeech	4.43(±0.06)
VITS	标贝中文	4.24(±0.06)
VITS	四川方言	4.11(±0.06)

评价结果显示，VITS 对各种数据集都有较好的合成效果，而 Tacotron2+HiFiGAN 在训练样本文本量少的情况下，表现不如标准的多样本训练集。可以说明，VITS 具有更强的迁移能力和鲁棒性。

5.8 本章总结

在这一章中，本文对 VITS 进行了与而 Tacotron2 的对比实验，并介绍了实验所用的 MFCC，MCD，MOS 等指标参数，在主观与客观层面验证了 VITS 模型的强大语音合成性能。我们可以总结出 VITS 的一些优势：

1）VAE 擅长捕捉句子整体的韵律特征，而 Flow 擅长重建音频的细节特征；将两者整合，进行多任务训练，实现参数与优势共享。

2）VITS 直接合成音频，实现真正意义的端到端语音合成，而不是阶段的序列到序列的合成；从而消除两个模型带来的 Gap 误差。

3）传统两个模型的 TTS 系统，GAN 训练通常只应用与声码器，而 VITS 中的 GAN 训练是全局的、对每个模块都有效。

VITS 和 Tacotron2 都是语音合成模型，但是它们的方法和目标略有不同。VITS 是一种基于迭代的语音合成模型，旨在提高语音质量和流畅性，而 Tacotron2 是一种基于注意力机制的文本到语音合成模型，旨在从文本生成连贯的语音信号。

下面是它们之间的具体比较:：

1）方法：Tacotron2 是一种基于注意力机制的文本到语音合成模型，而 VITS

是一种基于迭代的语音合成模型。Tacotron2 使用一个编码器将文本转换为中间表示，然后使用一个解码器生成语音信号；VITS 则直接使用一个深度神经网络生成语音信号。

2）目标：Tacotron2 的目标是从文本生成连贯的语音信号，而 VITS 的目标是提高语音合成的质量和流畅性。

3）训练：Tacotron2 的训练需要大量的标注语音数据和对应的文本数据，而 VITS 可以通过无监督学习或少量监督学习进行训练。

4）性能：Tacotron2 在一些公开数据集上的语音合成质量已经达到了较高水平，但它的语音合成速度相对较慢。而 VITS 则可以生成高质量的语音信号，并且速度较快，但需要更多的计算资源和时间进行训练。

第六章 全文内容总结与展望

6.1 全文内容总结

在本文中，我首先梳理了 TTS 模型的研究现状及发展态势，以及传统 TTS 模型和端到端 TTS 模型的架构。在此基础上，介绍了 VITS 这种端到端的语音合成模型及其结构原理，然后设计实验，制作四川方言数据集，并将四川方言数据集迁移到该模型中进行训练，并对合成语音进行了详细的评估，使用了 MOS 来主观评价和 MCD 来客观评价。同时，我们采用对比实验，将 VITS 和 Tacotron2+HiFiGAN 语音合成模型进行对比，进一步说明了 VITS 模型合成语音的高表现力，高迁移能力和流畅性。

6.2 展望

对于 VITS 而言，其生成语音的流畅性已经有目共睹，同时，我们也注意到了 VITS 的一些问题，比如，VITS 虽然通过随机时长预测器解决了一对多（one-to-many）的问题，但是从合成结果来看，因为对韵律，语调等特征是统一建模的，没有进行拆分细粒度，因此针对方言之类的语调丰富的语言，其语调建模的学习的比较平均化，针对风格发音人来说，整体合成效果来看无法得到较好的语调，比较单一，语调较平。无法生成类似真人说话的情绪波动。而且，我们注意到有 MCD 的值较大，在传统语音识别项目中，通常认为 MCD 要小于 8 才能够被机器识别，我认为问题在与随机时长预测导致每一帧无法与原始音频的每一帧完美对应。而一般来说，通过语音转换（Voice Conversion）可以有效解决帧对齐的问题。所以，如何将该模型应用于语音转换也是目前的一大目标。目前，已经有使用 VITS 制作的 so-vits-svc 项目可以实现语音转换和歌声转换。同时，由于 VITS 采用了 CVAE 来判别语句中各个字符对应的音素，同样，CVAE 也可以作为一种情感生成模型，在训练集中添加相应的情感标签，对生成的语音添加情感向量，这样在合成的过程中，我们就可以制定合成语音的情感，以此合成出更具有情绪波动的音频。此外，我们也可以进行多尺度的韵律建模，在细粒度和粗粒度的音素上进行建模，例如同时考虑各字符的音素长度和整个语句的音素长度，以期望合成出更加自然且富有情感的语音。

致 谢

在攻读学士学位期间，首先衷心感谢我的导师史创教授，对我的毕业设计进行了详尽的指导，在文章中出现任何问题时及时的反馈给我们信息，让我们能够尽快的打磨好一篇优秀的论文。同时我也要感谢一路上帮助过我的同学和辅导员，给了我许多经验上的指导和贴心的关怀。此外，感谢我的父母二十余年的抚养，没有他们的辛苦劳累，就没有今天的我，由衷地谢谢我的爸爸妈妈。

参考文献

[1] J. Kim, J. Kong, J. Son. Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech[C]. Proceedings of the 38th International Conference on Machine Learning, 2021, 5530-5540

[2] E. Hoogeboom, J. W. Peters, R. van den Berg, et al. Integer discrete flows and lossless compression[M]. Red Hook, NY, USA: Curran Associates Inc., 2019

[3] J. Shen, R. Pang, R. J. Weiss, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions[C]. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, 4779-4783

[4] A. Oord, S. Dieleman, H. Zen, et al. Wavenet: A generative model for raw audio[J]. , 2016, :

[5] Y. Ren, C. Hu, X. Tan, et al. Fastspeech 2: Fast and high-quality end-to-end text to speech[J]. ArXiv, 2020, abs/2006.04558:

[6] J. Gou, B. Yu, S. J. Maybank, et al. Knowledge distillation: A survey[J]. Int. J. Comput. Vision, 2021, 129(6): 1789 1819

[7] M. Yang, Z. Wang, Z. Chi, et al. Wavegan: Frequency-aware gan for high-fidelity few-shot image generation[J]. ArXiv, 2022, :

[8] D. P. Kingma, M. Welling. An introduction to variational autoencoders[J]. Found. Trends Mach. Learn., 2019, 12(4): 307 392

[9] I. Kobyzev, S. J. Prince, M. A. Brubaker. Normalizing flows: An introduction and review of current methods[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(11): 3964-3979

[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, et al. Generative adversarial networks[J]. Commun. ACM, 2020, 63(11): 139 144

[11] S. Ramchandran, G. Tikhonov, O. Lönnroth, et al. Learning conditional variational autoencoders with missing covariates[J]. ArXiv, 2022, abs/2203.01218:

[12] D. P. Kingma, M. Welling. Auto-encoding variational bayes[J]. CoRR, 2013, abs/1312.6114:

[13] U. Michelucci. An introduction to autoencoders[J]. , 2022, :

[14] J. Kim, S. Kim, J. Kong, et al. Glow-tts: A generative flow for text-to-speech via monotonic alignment search[J]. ArXiv, 2020, :

[15] J. Kong, J. Kim, J. Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis[C]. , Red Hook, NY, USA, 2020,

[16] A. Vaswani, N. Shazeer, N. Parmar, et al. Attention is all you need[C]. Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2017, 6000 6010

[17] K. He, X. Zhang, S. Ren, et al. Deep residual learning for image recognition[C]. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, 770-778

[18] L. Kaiser, A. N. Gomez, F. Chollet. Depthwise separable convolutions for neural machine translation[J]. ArXiv, 2017, :

[19] H. Liang, X. Hou, L. Shen. Ssflow: Style-guided neural spline flows for face image manipulation[C]. Proceedings of the 29th ACM International Conference on Multimedia, New York, NY, USA, 2021, 79 87

[20] J. Ho, X. Chen, A. Srinivas, et al. Flow++: Improving flow-based generative models with variational dequantization and architecture design[J]. ArXiv, 2019, :

[21] C. Chadebec, S. Allassonnière. Data augmentation with variational autoencoders and manifold sampling[C]. Deep Generative Models, and Data Augmentation, Labelling, and Imperfections: First Workshop, DGM4MICCAI 2021, and First Workshop, DALI 2021, Held in Conjunction with MICCAI 2021, Strasbourg, France, October 1, 2021, Proceedings, Berlin, Heidelberg, 2021, 184 192

[22] I. Loshchilov, F. Hutter. Decoupled weight decay regularization[C]. International Conference on Learning Representations, 2019,

[23] P. Micikevicius, S. Narang, J. Alben, et al. Mixed precision training[J]. , 2018, ():

[24] J. Chorowski, D. Bahdanau, D. Serdyuk, et al. Attention-based models for speech recognition[C]. Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, Cambridge, MA, USA, 2015, 577 585

[25] M. MORISE, F. YOKOMORI, K. OZAWA. World: A vocoder-based high-quality speech synthesis system for real-time applications[J]. IEICE Transactions on Information and Systems, 2016, E99.D(7): 1877-1884

[26] Z. K. Abdul, A. K. Al-Talabani. Mel frequency cepstral coefficient and its applications: A review[J]. IEEE Access, 2022, 10(): 122136-122158

[27] W. contributors. Mean opinion score[J]. https://en.wikipedia.org/w/index.php?, 2022, ():

[28] J. Sueur. Mel-frequency cepstral and linear predictive coefficients[M]. Cham: Springer International Publishing, 2018, 381-398

外文资料原文

1 Conditional Variational Autoencoder with Adversarial Learning for End-to-End Text-to-Speech

Several recent end-to-end text-to-speech (TTS) models enabling single-stage training and parallel sampling have been proposed, but their sample quality does not match that of two-stage TTS systems. In this work, we present a parallel end to-end TTS method that generates more natural sounding audio than current two-stage models. Our method adopts variational inference augmented with normalizing flows and an adversarialtraining process, which improves the expressive power of generative modeling. We also propose astochastic duration predictor to synthesize speech with diverse rhythms from input text. With the uncertainty modeling over latent variables and the stochastic duration predictor, our method expresses the natural one-to-many relationship in which a text input can be spoken in multiple ways with different pitches and rhythms. A subjective human evaluation (mean opinion score, or MOS) on the LJ Speech, a single speaker dataset, shows that our method outperforms the best publicly available TTS systems and achieves a MOS comparable to ground truth.

Text-to-speech (TTS) systems synthesize raw speech waveforms from given text through several components. With the rapid development of deep neural networks, TTS system pipelines have been simplified to two-stage generative modeling apart from text preprocessing such as text normalization and phonemization. The first stage is to produce intermediate speech representations such as melspectrograms (Shen et al., 2018) or linguistic features (Oordet al., 2016) from the preprocessed text, and the second stage is to generate raw waveforms conditioned on the in termediate representations (Oord et al., 2016; Kalchbrenner et al., 2018). Models at each of the two-stage pipelines have been developed independently.

Neural network-based autoregressive TTS systems have shown the capability of synthesizing realistic speech (Shen et al., 2018; Li et al., 2019), but their sequential generative process makes it difficult to fully utilize modern parallel processors. To overcome this limitation and improve synthesis speed, several non-autoregressive methods have been proposed. In the text-to-spectrogram generation step, extracting attention maps from pretrained autoregressive teacher networks (Ren et al., 2019; Peng et al., 2020) is attempted to decrease the difficulty of learning alignments between text and spectrograms. More recently, likelihood-based methods further eliminate the dependency on external aligners by estimating or learning alignments that maximize the likelihood of target mel-spectrograms (Zeng et al., 2020; Miao et al., 2020; Kim et al., 2020). Meanwhile, generative adversarial networks (GANs) (Goodfellow et al., 2014) have been explored in second stage models. GAN-based feed-forward networks with multiple discriminators, each distinguishing samples at different scales or periods, achieve high-quality raw waveform synthesis (Kumar et al., 2019; Bi´nkowski et al., 2019; Kong et al., 2020). Despite the progress of parallel TTS systems, two-stage pipelines remain problematic because they require sequential training or fine-tuning (Shen et al., 2018; Weiss et al.,2020) for high-quality production wherein latter stage models are trained with the generated samples of earlier stage models. In addition, their dependency on predefined intermediate features precludes applying learned hidden representations to obtain further improvements in performance. Recently, several works, i.e., FastSpeech 2s (Ren et al., 2021) and EATS (Donahue et al., 2021), have proposed efficient end-to-end training methods such as training over short audio clips rather than entire waveforms, leveraging a mel-spectrogram decoder to aid text representation learning,and designing a specialized spectrogram loss to relax length mismatch between target and generated speech. However, despite potentially improving performance by utilizing the learned representations, their synthesis quality lags behind two-stage systems.

In this work, we present a parallel end-to-end TTS method that generates more natural sounding audio than current two-stage models. Using a variational autoencoder (VAE) (Kingma & Welling, 2014), we connect two modules of TTS systems through latent variables to enable efficient end-to-end learning. To improve the expressive power of our method so that high-quality speech waveforms can be synthesized, we apply normalizing flows to our conditional prior distribution and adversarial training on the waveform domain. In addition to generating fine-grained audio, it is important for TTS systems to express the one-to-many relationship in which text input can be spoken in multiple ways with different variations (e.g., pitch and duration). To tackle the one-to-many problem, we also propose a stochastic duration predictor to synthesize speech with diverse rhythms from input text. With the uncertainty modeling over latent variables and the stochastic duration predictor, our method captures speech variations that cannot be represented by text.

Our method obtains more natural sounding speech and higher sampling efficiency than the best publicly available TTS system, Glow-TTS (Kim et al., 2020) with HiFiGAN (Kong et al., 2020). We make both our demo page and source-code publicly available.

外文资料译文

1 基于变分推理和对抗性机械学习的端到端文语转换模型

最近提出了几个能够进行单阶段训练和并行采样的端到端文本到语音 (TTS) 模型，但它们的采样质量与两阶段的 TTS 系统不相匹配。在这项工作中，我们提出了一种并行的端到端 TTS 方法，它比目前的两阶段模型产生更自然的声音。我们的方法采用了变异推理，并辅以归一化流和对抗性训练过程，从而提高了生成式建模的表达能力。我们还提出了一个随机的持续时间预测器，以便从输入的文本中合成具有不同节奏的语音。通过对潜在变量的不确定性建模和随机持续时间预测器，我们的方法体现了自然的一对多关系，即一个文本输入可字鞍蔼啊以用不同的音高和节奏以多种方式说话。在 LJspeech (一个单人说话的数据集) 上进行的人类主观评价 (平均意见得分，或 MOS) 表明，我们的方法优于最好的公开可用的 TTS 系统，并实现了与基础事实相类似的 MOS。

文本到语音 (TTS) 系统通过几个组件从给定的文本中合成原始语音波形。随着深度神经网络的快速发展 TTS 系统的管道已经简化为两阶段的生成模型，除了文本预处理，如文本规范化和音素化。第一阶段是产生中间的语音表征，如旋律谱图，或语言学特征，从预处理的文本中、第二阶段是生成以中间表征为条件的原始波形。两阶段管道中的每个模型都是独立开发的。

基于神经网络的自回归 TTS 系统已经显示出合成真实语音的能力，但其顺序生成过程使其难以充分利用现代并行处理器。为了克服这一限制并提高合成速度，已经提出了几种非自回归方法。在文本到频谱的生成步骤中，试图从预先训练好的自回归教师网中提取注意力图，以减少文本和频谱之间学习对齐的难度。最近，基于可能性的方法进一步消除了对外部对准器的依赖，通过估计或学习最大化目标旋律-谱图的可能性的对准。同时，生成式广告网络 (GANs) 已经在第二阶段模型中进行了探索。基于 GAN 的前馈网络有多个判别器，每个判别器区分不同尺度或周期的样本，实现高质量的原始波形合成。

尽管并行 TTS 系统取得了进展，但两阶段管道仍然存在问题，因为它们需要连续训练或微调以实现高质量的生产，其中后一阶段的模型是用前一阶段的模型生成的样本进行训练。此外，他们对预定义的中间特征的依赖排除了应用学习的隐藏代表来获得性能的进一步改善。最近些作品，即 FastSpeech 2 和 EATS，提出了高效的端到端训练方法，如通过短的音频片段而不是整个波形进行训练，利用旋律谱解码器来帮助文本表示学习。

并设计一个专门的频谱图损失，以放松长度-目标和生成的语音之间不匹配。然而、尽管通过利用学习到的表征可能会提高性能，但其合成质量却落后于两阶段系统。

在这项工作中，我们提出了一种并行的端到端 TTS 方法，与目前的两阶段模型相比，它能产生更自然的声音。使用变异自动编码器 (VAE)，我们通过潜在变量连接 TTS 系统的两个模块，以实现高效的端到端学习。为了提高我们方法的表达能力，以便能够合成高质量的语音波形，我们对我们的条件先验分布和波形域的对抗性训练应用了归一化流。除了生成细粒度的音频，对于 TTS 系统来说，表达一对多的关系也很重要，在这种关系中，文本输入可以通过不同的变化 (如音调和持续时间) 以多种方式说出。为了解决一对多的问题，我们还提出了一个随机的二进制预测器，以便从输入文本中合成具有不同节奏的语音。通过对潜在变量的不确定性建模和随机持续时间预测器，我们的方法可以捕捉到无法用文本表示的语音变化。

我们的方法比最好的公开可用的 TTS 系统 Glow-TTS 和 HiFi-GAN 获得了更自然的语音和更高的采样效率。我们公开了我们的演示页面和源代码。