# Finding Sparse Subnetworks in Self-Supervised Speech Recognition and Speech Synthesis

by

Cheng-I Jeff Lai

B.S., Johns Hopkins University (2018)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022



Author . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
James R. Glass
Senior Research Scientist
Thesis Supervisor

Accepted by . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

# Finding Sparse Subnetworks in Self-Supervised Speech Recognition and Speech Synthesis

by

Cheng-I Jeff Lai

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

## Abstract

The modern paradigm in speech processing has demonstrated the importance of scale and compute for end-to-end speech recognition and synthesis. For instance, state-of-the-art self-supervised speech representation learning models typically consists of more than 300M model parameters and being trained on 24 GPUs. While such a paradigm has proven to be effective in certain offline settings, it remains unclear the extent to which it can be extended to online and small-device scenarios.

This thesis is a step toward making advanced speech processing models more parameter-efficient. We aim to answer the following: do sparse subnetworks exist in modern speech processing models, and if so, how can we discover them efficiently? The key contribution is a new pruning algorithm termed Prune-Adjust-Re-Prune (`PARP`), that discovers sparse subnetworks efficiently. `PARP` is inspired by our observation that subnetworks pruned for pre-training tasks need merely a slight adjustment to achieve a sizeable performance boost in downstream ASR tasks. We first demonstrate its effectiveness for self-supervised ASR in various low-resource settings. In particular, extensive experiments verify (1) sparse subnetworks exist in mono-lingual/multi-lingual pre-trained self-supervised learning representations, and (2) the computational advantage and performance gain of `PARP` over baseline pruning methods.

In the second study, we extend `PARP` to end-to-end TTS, including both spectrogram prediction networks and vocoders. We thoroughly investigate the tradeoffs between sparsity and its subsequent effects on synthetic speech. The findings suggest that not only are end-to-end TTS models highly prunable, but also, perhaps surprisingly, pruned TTS models can produce synthetic speech with equal or higher naturalness and intelligibility, with similar prosody.

Thesis Supervisor: James R. Glass
Title: Senior Research Scientist

# Acknowledgments

Always grateful to those who have helped me along the way.

To my undergraduate research mentors at JHU and Edinburgh: Najim, Jesus, Simon, Korin, Alberto, and Hynek – thank you for being patient with me, teaching me so much about the subject, and giving me the level of trust I still don't think I deserved.

To other members from CLSP and CSTR: thank you for the many conversations we had over lunch, coffee break, dinner, or afternoon walks. Many of you were my research idols and inspirations, and shaped my research outlook early on.

To Junichi and members from NII: thank you for teaching me so much about synthesis, offering me a stay in the heart of Tokyo in summer 2019, and teaching me about both Japanese culture and your research principle. It is incredible that the one month visit turned into a long-term collaboration, with 5 co-authored papers thus far.

To Hao, Dave, Sameer, Wei-Ning, Yu-An, and Alex: thank you for distilling your passion on unsupervised speech processing into me, which later becomes the theme of my PhD research. Your past research work has already become the stepping stone toward my PhD goal. Your scholarly establishments have always been one of the bars I am pursuing after in research. Your research connections have indirectly helped me land internship opportunities at AWS AI and Meta AI Research.

To Daniel Li, Jin, and others from AWS: thank you for offering my first industry internship in summer 2020. I learned about SLU and end-to-end ASR, and generally, what applied research is like in industry. The internship lead to an ongoing collaboration with Hung-Yi and many others on self-supervised learning in speech since 2021, which further opened up even more research and learning opportunities later on.

To Yang, Shiyu, Kaizhi, and others from MIT-IBM: thank you for teaching me about many aspects of research, from delving deep into a research topic, making posters, planning a career in academic research, to properly addressing rebuttals. I remember our weekly zoom meetings in Spring 2021, and distinctly Yang's confidence in our NeurIPS submission, which lays the foundation for this SM thesis.

To friends at CSAIL, ROCSA, and my roommates: thank you for tolerating many of my foolishness. Your companionship makes this long journey so much more enjoyable.

To mom and dad: thank you for all the sacrifice. I love you.

To Vivian and Daniel: I spent the majority of my time with you in my first three years of grad school. There is statistics about how terrifying PhD is – self-doubt, insecurities about money and the future, peer-pressure, etc. – I have experienced it all, and unfortunately you are among the very few who knew about it. However, despite the seemingly overwhelming negativity at times, I only recall the better part of what we went through. The peaceful moments at Tang Hall, KBL, and Meridian during the pandemic, and the crazy workations in Miami, Puerto Rico, Vegas, Hawaii, Fremont, and Mexico would stay on with me forever.

To Jim: thank you for admitting me into MIT, giving me the time and space to explore, and being supportive to my many careless mistakes and non-research decisions. Your research professionalism and passion in unsupervised speech processing have a profound impact in my career growth ever since I joined SLS. I am looking forward to the second half of the PhD research.

My SM thesis is dedicated to my grandparents, who passed away in summer 2020. They generously supported my college education, and made all of this possible.

The thesis is supported by the Merrill Lynch Fellowship and MIT-IBM AI Lab.

# Bibliographic Note

The work presented in this thesis has previously appeared in peer-reviewed publications. Chapter 3 was published in Lai et al. 2021 at NeurIPS 2021. Chapter 4 was published in Lai et al. 2022 at ICASSP 2022.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Over the last decade, we have seen significant advancement in end-to-end and self-supervised learning in spoken language processing. One key lesson that emerges over time is the importance of model scaling, regardless of training objectives, supervision, or architecture, in order to attain state-of-the-art results. While models with increasing large number of parameters has proven to be effective in various benchmarks and in-house data sets, there is much space for improvement when it comes to extending the same technology to more limited settings. Given that there is little work in parameter-efficiency for speech, this thesis work focuses on developing insights into finding sparse subnetwork in modern speech processing models, with the ultimate goal of reducing the training and inference requirement of these state-of-the-art models.

## 1.1 Contributions

The main contributions of this thesis center around a novel pruning algorithm termed `PARP`. We conduct extensive `PARP` and baseline (`OMP` and `IMP`) pruning experiments on low-resource ASR with mono-lingual (pre-trained wav2vec 2.0 (Baevski et al., 2020)) and cross-lingual (pre-trained XLSR-53 (Conneau et al., 2020)) transfer. `PARP` finds significantly superior speech SSL subnetworks for low-resource ASR, while only requiring a single pass of downstream ASR finetuning. We then extends `PARP` to synthesis, with the intention of not only reducing architectural complexity for end-to-end TTS, but also demonstrating the surprising efficacy and simplicity of pruning in contrast to prior TTS efficiency work. The summary of contributions are:

- We show that sparse subnetworks exist in pre-trained speech SSL when finetuned for low-resource ASR. In addition, `PARP` achieves superior results to `OMP` and `IMP` across all sparsities, amount of finetuning supervision, pre-trained model scale, and downstream spoken languages. Specifically, on Librispeech 10min without LM decoding, `PARP` discovers subnetworks from wav2vec 2.0 with an absolute 10.9%/12.6% WER decrease compared to the full model, without modifying the finetuning hyper-parameters or objective.

- `PARP` minimizes phone recognition error increases in cross-lingual mask transfer, where a subnetwork pruned for ASR in one spoken language is adapted for ASR in another language. `PARP` can also be applied to efficient multi-lingual subnetwork discovery for 10 spoken languages.

- We also demonstrate `PARP`'s effectiveness on pre-trained BERT/XLNet, mitigating the cross-task performance degradation reported in BERT-Ticket (Chen et al., 2020b).

- We present the first comprehensive study on pruning end-to-end acoustic models (Transformer-TTS (Li et al., 2019), Tacotron2 (Shen et al., 2018)) and vocoders (Parallel WaveGAN (Yamamoto et al., 2020)) with `PARP`.

- We show that end-to-end TTS models are over-parameterized. Pruned models produce speech with similar levels of naturalness, intelligibility, and prosody to that of unpruned models.

## 1.2 Thesis Outline

The thesis is composed of three main chapters. In Chapter 2, we first lay the foundation of neural network pruning, and some hurdles when it is naively applied to speech processing. In Chapter 3, we formulate the proposed algorithm `PARP` and its application to pre-trained self-supervised representations in low-resource speech recognition (ASR) settings. In Chapter 4, we further extend `PARP` to end-to-end speech synthesis (TTS), gaining insights into the effect of sparsity in synthesis naturalness, intelligibility, and prosody. Chapter 5 concludes with a brief summary of the thesis.

# Chapter 2

# Sparse Subnetwork Discovery in Neural Networks

## 2.1 Introduction

Neural network pruning (LeCun et al., 1990; Hassibi and Stork, 1993; Han et al., 2015; Li et al., 2016), as well as the more recently proposed Lottery Ticket Hypothesis (LTH) (Frankle and Carbin, 2018), suggests the existence of sparse subnetworks in pre-trained neural networks. According to LTH, there exists sparse subnetworks that can achieve the same or *even better* accuracy than the original dense network. Such phenomena have been successfully observed in various domains: Natural Language Processing (NLP) (Yu et al., 2019; Chen et al., 2020b; Prasanna et al., 2020; Movva and Zhao, 2020), Computer Vision (CV) (Chen et al., 2020a; Girish et al., 2020), and many others. All finding sparse subnetworks with comparable or better performance than the dense network. Given the lack of similar studies on pruning speech processing models, we intend to fill this gap by finding sparse subnetworks in pre-trained Automatic Speech Recognition(ASR) and Speech Synthesis (TTS) models.

## 2.2 Sparse Subnetwork Discovery in Speech

### 2.2.1 Formulation

Consider a sequence-to-sequence learning problem, where $\boldsymbol{X}$ and $\boldsymbol{Y}$ represent the input and output sequences respectively. For ASR, $\boldsymbol{X}$ is waveforms and $\boldsymbol{Y}$ is character/phone sequences; for a TTS acoustic model, $\boldsymbol{X}$ is character/phone sequences and $\boldsymbol{Y}$ is spectrogram sequences; for a vocoder, $\boldsymbol{X}$ is spectrogram sequences and $\boldsymbol{Y}$ is waveforms. A mapping function $f(\boldsymbol{X};\theta)$ parametrized by a neural network is learned, where $\theta \in \mathcal{R}^d$ represents the network parameters and $d$ represents the number of parameters. Sequence-level log-likelihood $\mathbb{E}\left[\ln P(\boldsymbol{Y} \mid \boldsymbol{X};\theta)\right]$ on target dataset $\mathcal{D}$ is maximized.

The goal of sparse subnetwork discovery is to find a subnetwork $m \odot \theta$, where $\odot$ is the element-wise product and a binary pruning mask $m \in \{0,1\}^d$ is applied on the model weights $\theta$. The ideal pruning method would learn $m$ at target sparsity such that $f(\boldsymbol{X}; m \odot \theta)$ achieves similar loss as $f(\boldsymbol{X};\theta)$ after training on $\mathcal{D}$.

### 2.2.2 Methods

**Unstructured Magnitude Pruning (`UMP`)** (Frankle and Carbin, 2018; Gale et al., 2019) sorts the model's weights according to their magnitudes across layers regardless of the network structure, and removes the smallest ones to meet a predefined sparsity level. Weights that are pruned out (specified by $m$) are zeroed out and do not receive gradient updates during training.

**One-Shot Magnitude Pruning (`OMP`)** (Frankle and Carbin, 2018; Gale et al., 2019) is based on `UMP` and assumes an initial model weight $\theta_0$ and a target dataset $\mathcal{D}$ are given. `OMP` can be described as:

1. Directly prune $\theta_0$ at target sparsity, and obtain an initial pruning mask $m_0$. Zero out weights in $\theta_0$ given by $m_0$.

2. Train $f(\boldsymbol{X}; m_0 \odot \theta_0)$ on $\mathcal{D}$ until convergence. Zeroed-out weights do not receive

gradient updates via backpropogation.

**Iterative Magnitude Pruning (`IMP`)** extends `OMP` to multiple iterations by updating $\theta_0$ with the finetuned model weight $\theta_D^*$ from Step 2.

### 2.2.3 Shortcomings

Directly applying the above pruning methods to speech processing suffers from two challenges. First, these methods applied to SOTA speech models, either ASR or TTS, is extremely time-consuming. `OMP` and `IMP` involve more than one round of training on $\mathcal{D}$ (c.f. Figure 2-1), and yet one-round of ASR or TTS training is prohibitively time-consuming and computationally demanding compared to NLP or CV[1]. The second challenge is that we do not observe *any* performance improvement of the subnetworks over the original dense network with `OMP` or `IMP`. Figure 3-2 shows the WER under low-resource scenarios of the subnetworks identified by `OMP` (purple line) and `IMP` (blue dashed line) at different sparsity levels. None of the sparsity levels achieves a visible drop in WER compared to the zero sparsity case, corresponding to the original dense network. These two challenges have prompted us to ask – do there exist sparse subnetworks within in SOTA speech processing models? Furthermore, how can we discover them efficiently?

[1]Standard wav2vec 2.0 finetuning setup (Baevski et al., 2020) on any Librispeech/Libri-light splits requires at least 50∼100 V100 hours, which is more than 50 times the computation cost for finetuning a pre-trained BERT on GLUE (Wang et al., 2018).

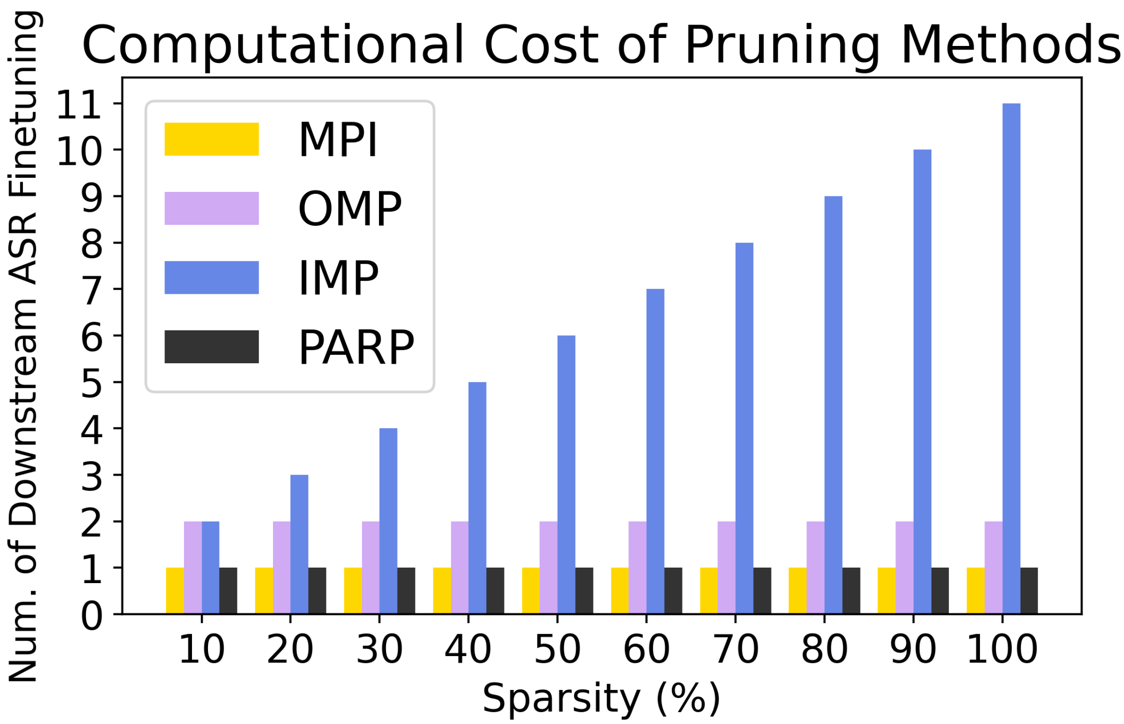


Figure 2-1: Number of ASR finetuning iterations needed (y-axis) versus target sparsities (x-axis) for *each* downstream task/language. Cross-referencing Figure 3-2 indicates that `IMP` requires linearly more compute to match the performance (either sparsity/WER) of `PARP`.

# Chapter 3

# Finding Sparse Subnetworks in Self-Supervised Speech Recognition

## 3.1 Introduction

For many low-resource spoken languages in the world, collecting large-scale transcribed corpora is very costly and sometimes infeasible. Inspired by efforts such as the IARPA BABEL program, Automatic Speech Recognition (ASR) trained without sufficient transcribed speech data has been a critical yet challenging research agenda in speech processing (Cui et al., 2013, 2014; Gales et al., 2014; Cui et al., 2015; Cho et al., 2018). Recently, Self-Supervised Speech Representation Learning (speech SSL) has emerged as a promising pathway toward solving low-resource ASR (Oord et al., 2018b; Chung et al., 2019; Wang et al., 2020; Baevski et al., 2020; Conneau et al., 2020; Zhang et al., 2020; Hsu et al., 2021c; Chung et al., 2021b). Speech SSL involves pre-training a speech representation module on large-scale *unlabelled* data with a self-supervised learning objective, followed by finetuning on a small amount of supervised transcriptions. Many recent studies have demonstrated the empirical successes of speech SSL on low-resource English and multi-lingual ASR, matching systems trained on fully-supervised settings (Baevski et al., 2020; Conneau et al., 2020; Zhang et al., 2020; Baevski et al., 2021; Zhang et al., 2021a). Prior research attempts, however, focus on pre-training objectives (Oord et al., 2018b; Chung et al., 2019; Wang et al., 2020; Liu et al., 2020a; Jiang et al., 2020; Liu et al., 2020b; Ling and Liu, 2020; Liu et al., 2021; Hsu et al., 2021c; Chorowski et al., 2021; Chung et al., 2021b; Chen et al., 2021d; Zhu et al., 2021), scaling up speech representation modules (Baevski et al., 2019b, 2020; Hsu et al., 2021a), pre-training data selections (Wang et al., 2021b; Hsu et al., 2021b; Wang et al., 2021a,d; Meng et al., 2021), or applications of pre-trained speech representations (Chung et al., 2018; Lai, 2019; Rivière et al., 2020; Chung et al., 2020; Lai et al., 2021a; Conneau et al., 2020; Maekaku et al., 2021; Yang et al., 2021; Lakhotia et al., 2021; Xu et al., 2021a; Wiesner et al., 2021; Gao et al., 2021; Baevski et al., 2021; Polyak et al., 2021; Kharitonov et al., 2021; Lee et al., 2021; Ao et al., 2021; Huang et al., 2021; Tseng et al., 2021; Chang et al., 2021; Cooper et al., 2021; Chen et al., 2021e). In this work, we aim to develop an orthogonal approach that is complementary to these existing speech SSL studies, that achieves 1) lower architectural complexity and 2) higher performance (lower WER) under the same low-resource ASR settings.

### 3.1.1 Background

As model scale (Synnaeve et al., 2019; Baevski et al., 2020; Han et al., 2020; Gulati et al., 2020; Yu et al., 2020; Pratap et al., 2020b,a; Yu et al., 2021; Chen et al., 2021c; You et al., 2021; Li et al., 2021a) and model pre-training (Baevski et al., 2020; Zhang et al., 2020; Conneau et al., 2020; Kong et al., 2020b; Jiang et al., 2020; Lai et al., 2021a; Hsu et al., 2021c; Xu et al., 2021b; Chan et al., 2021; Kanda et al., 2021; Sanabria et al., 2021; Saeed et al., 2021; Ng et al., 2021; Polyak et al., 2021; Wang et al., 2021c) have become the two essential ingredients for obtaining SOTA performance in ASR and other speech tasks, applying and developing various forms of memory-efficient algorithms, such as network pruning, to these large-scale pre-trained models will predictably soon become an indispensable research endeavor. Early work on ASR pruning can be dated back to pruning decoding search spaces (Abdou and Scordilis, 2004; Pylkkönen, 2005; Siivola et al., 2007; He et al., 2014; Xu et al., 2018; Zhang et al., 2021b) and HMM state space (Van Hamme and Van Aelten, 1996). Since the seminal work of Yu et al. (Yu et al., 2012), ASR pruning has focused primarily on end-to-end network architectures: (Shangguan et al., 2019; Wu et al., 2021) applied pruning and quantization to LSTM-based RNN-Transducers, (Panchapagesan et al., 2021) applied knowledge distillation to Conformer-based RNN-Transducers, (Venkatesh et al., 2021; Shi et al., 2021; Li et al., 2021b) designed efficient architecture/mechanisms for LSTM, Transformer, Conformer-based ASR models, (Narang et al., 2017) applied pruning to Deep Speech, (Braun and Liu, 2019) introduced SNR-based probabilistic pruning on LSTM-based CTC model, (Gao et al., 2020) proposed entropy-regularizer for LSTM-based ASR model, (Xue et al., 2013; Povey et al., 2018) applied SVD on ASR models' weight matrices. We emphasize that our work is the first on pruning large self-supervised pre-trained models for low-resource and multi-lingual ASR. In addition, to our knowledge, none of the prior speech pruning work demonstrated the pruned models attain superior performance than its original counterpart.

### 3.1.2 Method Overview

We propose a magnitude-based unstructured pruning method (Gale et al., 2019; Blalock et al., 2020), termed Prune-Adjust-Re-Prune (`PARP`), for discovering sparse subnetworks within pre-trained speech SSL. `PARP` consists of the following two steps:

1. Directly prune the SSL pre-trained model at target sparsity, and obtain an initial subnetwork and an initial pruning mask.

2. Finetune the initial subnetwork on target downstream task/language. During finetuning, zero out the pruned weights specified by the pruning mask, but allow the weights be updated by gradient descent during backpropogation. After a few number of model updates, re-prune the updated subnetwork at target sparsity again.

Step 1 provides an initial subnetwork that is agnostic to the downstream task, and Step 2 makes learnable adjustments by reviving pruned out weights. A formal and generalized description and its extension are introduced in Section 3.3. Different from pruning methods in (Han et al., 2015; Frankle and Carbin, 2018), `PARP` allows pruned-out weights to be revived during finetuning. Although such a high-level idea was introduced in (Guo et al., 2016), we provide an alternative insight: despite its flexibility, Step 2 only makes **minimal adjustment** to the initial subnetwork, and obtaining a good initial subnetwork in Step 1 is the key. We empirically show that *any* task-agnostic subnetwork surprisingly provides a good basis for Step 2, suggesting that the initial subnetwork can be cheaply obtained either from a readily available task/language or directly pruning the pre-trained SSL model itself. In addition, this observation allows us to perform cross-lingual pruning (mask transfer) experiments, where the initial subnetwork is obtained via a different language other than the target language.

## 3.2 Preliminaries

Consider the low-resource ASR problem, where there is only a small transcribed training set $(x, y) \in \mathcal{D}_l$. Here $x$ represents input audio, and $y$ represents output transcription. Subscript $l \in \{1, 2, \cdots\}$ represents the downstream spoken language identity. Because of the small dataset size, empirical risk minimization generally does not yield good results. Speech SSL instead assumes there is a much larger unannotated dataset $x \in \mathcal{D}_0$. SSL pre-trains a neural network $f(x; \theta)$, where $\theta \in \mathcal{R}^d$ represents the network parameters and $d$ represents the number of parameters, on some self-supervised objective, and obtains the pre-trained weights $\theta_0$. $f(x; \theta_0)$ is then finetuned on downstream ASR tasks specified by a downstream loss $\mathcal{L}_l(\theta)$, such as CTC, and evaluated on target dataset $\mathcal{D}_l$.

Our goal is to discover a subnetwork that minimizes downstream ASR WER on $\mathcal{D}_l$. Formally, denote $m \in \{0, 1\}^d$, as a binary pruning mask for the pre-trained weights $\theta_0$, and $\theta^l$ as the finetuned weights on $\mathcal{D}_l$. The ideal pruning method should learn $(m, \theta^l)$, such that the subnetwork $f(x; m \odot \theta^l)$ (where $\odot$ is element-wise product) achieves minimal finetuning $\mathcal{L}_l(\theta)$ loss on $\mathcal{D}_l$.

### 3.2.1 Pruning Targets and Settings

We adopted pre-trained speech SSL `wav2vec2` and `xlsr` for the pre-trained initialization $\theta_0$.

**wav2vec 2.0** We took wav2vec 2.0 base (`wav2vec2-base`) and large (`wav2vec2-large`) pre-trained on Librispeech 960 hours (Baevski et al., 2020). During finetuning, a task specific linear layer is added on top of `wav2vec2` and jointly finetuned with CTC loss.

**XLSR-53** (`xlsr`) shares the same architecture, pre-training and finetuning objectives as `wav2vec2-large`. `xlsr` is pre-trained on 53 languages sampled from CommonVoice, BABEL, and Multilingual LibriSpeech, totaling for 56k hours of multi-lingual speech data.

We consider three settings where `wav2vec2` and `xlsr` are used as the basis for low-resource ASR:

**LSR: Low-Resource English ASR.** Mono-lingual pre-training and finetuning – an English pre-trained speech SSL such as `wav2vec2` is finetuned for low-resource English ASR.

**H2L: High-to-Low Resource Transfer for Multi-lingual ASR.** Mono-lingual pre-training and multi-lingual finetuning – a speech SSL pre-trained on a high-resource language such as English is finetuned for low-resource multi-lingual ASR.

**CSR: Cross-lingual Transfer for Multi-lingual ASR.** Multi-lingual pre-training and finetuning – a cross-lingual pretrained speech SSL such as `xlsr` is finetuned for low-resource multi-lingual ASR.

### 3.2.2 Subnetwork Discovery in Pre-trained SSL

One obvious solution to the aforementioned problem is to directly apply pruning with rewinding to $\theta_0$, which has been successfully applied to pre-trained BERT (Chen et al., 2020b) and SimCLR (Chen et al., 2020a). All pruning methods, including our proposed `PARP`, are based on Unstructured Magnitude Pruning (`UMP`) (Frankle and Carbin, 2018; Gale et al., 2019), where weights of the lowest magnitudes are pruned out regardless of the network structure to meet the target sparsity level. We introduce four pruning baselines below, and we also provide results with Random Pruning (`RP`) (Frankle and Carbin, 2018; Gale et al., 2019; Chen et al., 2020b), where weights in $\theta_0$ are randomly eliminated.

**Task-Aware Subnetwork Discovery** is pruning with target dataset $D_l$ seen in advance, including One-Shot Magnitude Pruning (`OMP`) and Iterative Magnitude Pruning (`IMP`). `OMP` is summarized as:

1. Finetune pretrained weights $\theta_0$ on target dataset $\mathcal{D}_l$ to get the finetuned weights $\theta^l$.

2. Apply `UMP` on $\theta^l$ and retrieve pruning mask $m$.

`IMP` breaks down the above subnetwork discovery phase into multiple iterations – in our case multiple downstream ASR finetunings. Each iteration itself is an `OMP` with a fraction of the target sparsity pruned. We follow the `IMP` implementation described in BERT-Ticket (Chen et al., 2020b), where each iteration prunes out 10% of the *remaining* weights. The main bottleneck for `OMP` and `IMP` is the computational cost, since multiple rounds of finetunings are required for subnetwork discovery.

**Task-Agnostic Subnetwork Discovery** refers to pruning without having seen $D_l$ nor $l$ in advance. One instance is applying `UMP` directly on $\theta_0$ without any downstream finetuning to retrieve $m$, referred to as Magnitude Pruning at Pre-trained Initailizations (`MPI`). Another case is pruning weights finetuned for a different language $t$, *i.e.* applying `UMP` on $\theta^t$ for the target language $l$; in our study, we refer to this as cross-lingual mask transfer. While these approaches do not require target task finetuning, the discovered subnetworks generally have worse performance than those from `OMP` or `IMP`.

The above methods are only for subnetwork discovery via applying pruning mask $m$ on $\theta_0$. The discovered subnetwork $f(x; m \odot \theta_0)$ needs another downstream finetuning to recover the pruning loss[1], *i.e.* finetune $f(x; m \odot \theta_0)$ on $D_l$.

[1]This step is referred to as subnetwork finetuning/re-training in the pruning literature (Liu et al., 2018; Renda et al., 2020; Blalock et al., 2020).

## 3.3 Proposed Method

In this section, we highlight our proposed pruning method, `PARP` (Section 3.3.1), its underlying intuition (Section 3.3.2), and an extension termed `PARP-P` (Section 3.3.3).

### 3.3.1 Algorithm

We formally describe `PARP` with the notations from Section 3.2. A visual overview of `PARP` is Figure 3-6.

**Algorithm 1** Prune-Adjust-Re-Prune (PARP) to target sparsity $s$

1: Assume there are $N$ model updates in target task/language $l$'s downstream finetuning.
2: Take a pre-trained SSL $f(x;\theta_0)$ model. Apply task-agnostic subnetwork discovery, such as `MPI`[2], at target sparsity $s$ to obtain initial subnetwork $f(x;m_0 \odot \theta_0)$. Set $m = m_0$ and variable $n_1 = 0$ .
3: **repeat**
4: Zero-out masked-out weights in $\theta_{n1}$ given by $m$. Lift up $m$ such that whole $\theta_{n1}$ is updatable.
5: Train $f(x;\theta_{n1})$ for $n$ model updates and obtain $f(x;\theta_{n2})$.
6: Apply `UMP` on $f(x;\theta_{n2})$ and adjust $m$ accordingly. The adjusted subnetwork is $f(x;m \odot \theta_{n2})$. Set variable $n_1 = n_2$.
7: **until** total model updates reach $N$.
8: Return finetuned subnetwork $f(x;m \odot \theta_N)$.

Empirically, we found the choice of $n$ has little impact. In contrast to `OMP`/`IMP`/`MPI`, `PARP` allows the pruned-out weights to take gradient descent updates. A side benefit of `PARP` is it jointly discovers and finetunes subnetwork in a single pass, instead of two or more in `OMP` and `IMP`.

### 3.3.2 Obtaining and Adjusting the Initial Subnetwork

`PARP` achieves superior or comparable pruning results as task-aware subnetwork discov-

[2] By default, `MPI` is used for obtaining the initial subnetwork for `PARP` and `PARP-P` unless specified otherwise.

ery, while inducing similar computational cost as task-agnostic subnetwork discovery. How does it get the best of both worlds? The key is the discovered subnetworks from task-aware and task-agnostic prunings have high, non-trivial overlaps in LSR, H2L, and CSR. We first define Intersection over Union (`IOU`) for quantifying subnetworks' (represented by their pruning masks $m^a$ and $m^b$) similarity:

$$\texttt{IOU}(m^a, m^b) \triangleq \frac{|(m^a = 1) \cap (m^b = 1)|}{|(m^a = 1) \cup (m^b = 1)|} \tag{3.1}$$

Take H2L and CSR for instance, Figure 3-1 visualizes language pairs' `OMP` pruning mask `IOU`s on `wav2vec2` and `xlsr`. Observe the high overlaps across all pairs, but also the high `IOU`s with the `MPI` masks (second to last row). We generalize these observations to the following:

**Observation 1** *For any sparsity, any amount of finetuning supervision, any pre-training model scale, and any downstream spoken languages, the non-zero ASR pruning masks obtained from task-agnostic subnetwork discovery has high* `IOU`*s with those obtained from task-aware subnetwork discovery.*

Observation 1 suggests that *any* task-agnostic subnetwork could sufficiently be a good initial subnetwork in `PARP` due to the high similarities. In the same instance for H2L and CSR, we could either take `MPI` on `wav2vec2` and `xlsr`, or take `OMP` on a different spoken language as the initial subnetworks. Similarly in LSR, we take `MPI` on `wav2vec2` as the initial subnetwork. The underlying message is – the initial subnetwork can be obtained cheaply, without target task finetuning.

Now, because of the high similarity, the initial subnetwork (represented by its pruning mask $m_0$) needed merely a slight adjustment for the target downstream task. While there are techniques such as dynamic mask adjustment (Guo et al., 2016), important weights pruning (Molchanov et al., 2019), and deep rewiring (Bellec et al., 2017), we provide an even simpler alternative suited for our setting. Instead of permanently removing the masked-out weights from the computation graph, `PARP` merely zeroes them out. Weights that are important for the downstream task (the "important weights") should emerge with gradient updates; those that are relatively irrelevant should decrease in magnitude, and thus be zero-outed at the end. Doing so circumvents the need of straight-through estimation or additional sparsity loss, see Table 1 of (Sanh et al., 2020).

### 3.3.3 PARP-Progressive (`PARP-P`)

An extension to `PARP` is `PARP-P`, where the second `P` stands for Progressive. In `PARP-P`, the initial subnetwork starts at a lower sparsity, and progressively prune up to the target sparsity $s$ in Step 2. The intuition is that despite Observation 1, *not any* subnetwork can be a good initial subnetwork, such as those obtained from `RP`, or those obtained at very high sparsities in `MPI`/`OMP`/`IMP`. We show later that `PARP-P` is especially effective in higher sparsity regions, e.g. 90% for LSR. Note that `PARP-P` has the same computational cost as `PARP`, and the only difference is the initial starting sparsity in Step 1.

## 3.4 Experiments and Analysis

### 3.4.1 Comparing `PARP`, `OMP`, and `IMP` on LSR, H2L, and CSR

We first investigate the existence of sparse subnetworks in speech SSL. Figure 3-2 shows the pruning results on LSR. Observe that subnetworks discovered by `PARP` and `PARP-P` can achieve 60∼80% sparsities with minimal degradation to the full models. The gap between `PARP` and other pruning methods also widens as sparsities increase. For instance, Table 3.1 compares `PARP` and `PARP-P` with `OMP` and `IMP` at 90% sparsity, and `PARP-P` has a 40% absolute WER reduction. In addition, observe the WER reduction with `PARP` in the low sparsity regions on the 10min split in Figure 3-2. The same effect is not seen with `OMP`, `IMP`, nor `MPI`. Table 3.2 compares the subnetworks discovered by `PARP` with the full `wav2vec2` and prior work on LSR under the same setting[3]. Surprisingly, the discovered subnetwork attains an absolute

[3] We underscore again that LM decoding/self-training are not included to isolate the effect of pruning.

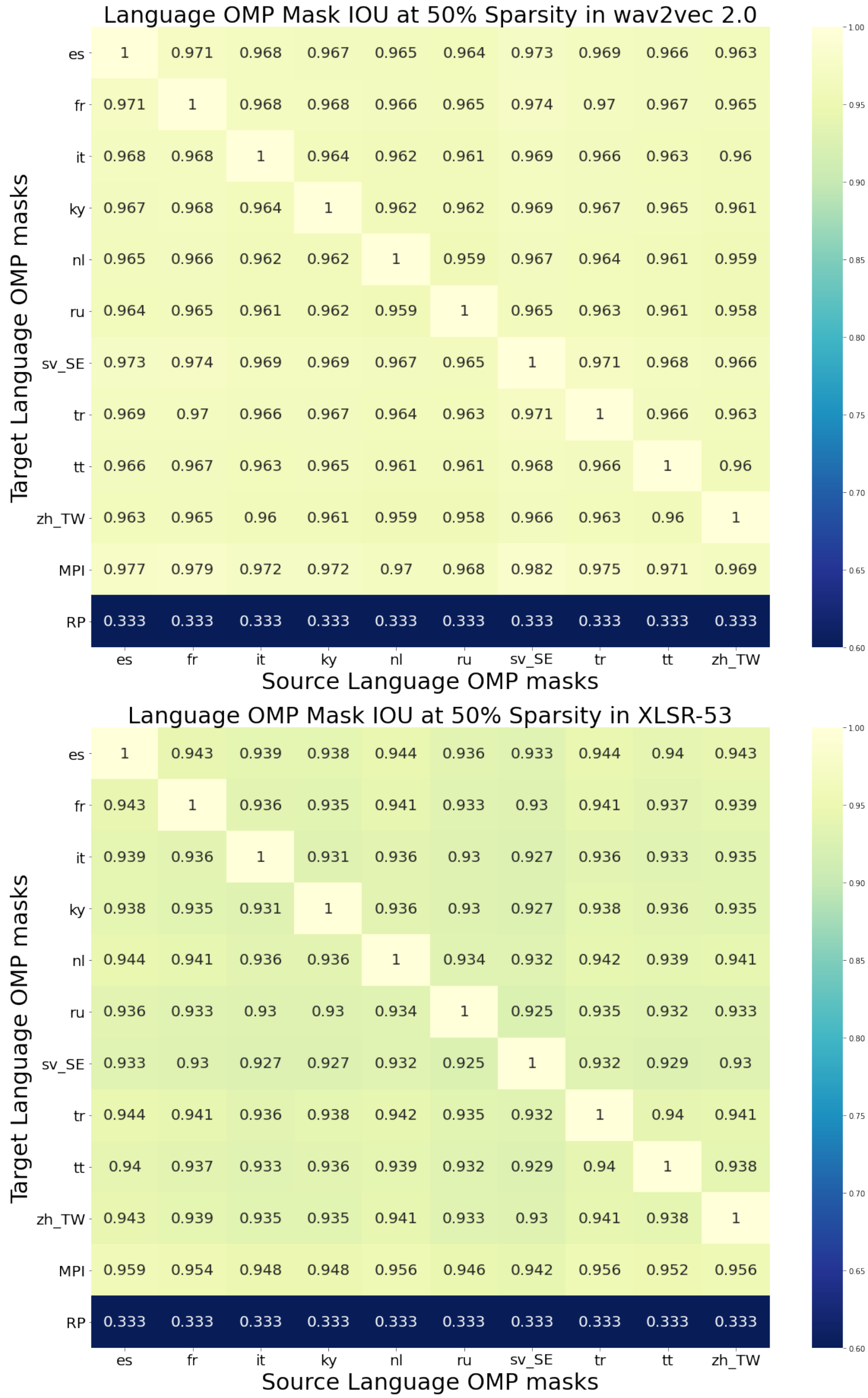


Figure 3-1: `IOU`s over all spoken language pairs' `OMP` pruning masks on finetuned `wav2vec2` and `xlsr`. Second to last row is the `IOU`s between `OMP` masks and the `MPI` masks from pre-trained `wav2vec2` and `xlsr`. Here, we show the `IOU`s at 50% sparsity.Surprisingly at any sparsities, there is a high, non-trivial (c.f. `RP` in the last row), similarity (>90%) between all spoken language `OMP` masks, as well as with the `MPI` masks.

10.9%/12.6% WER reduction over the full `wav2vec2-large`. We hypothesize that the performance gains are attributed to pruning out generic, unnecessary weights while preserving important weights, which facilitates training convergence. In other words, `PARP` provides additional regularization effects to downstream finetuning. We also examined the effectiveness of `IMP` with different rewinding starting points as studied in (Frankle et al., 2020; Renda et al., 2020), and found rewinding initializations bear minimal effect on downstream ASR.

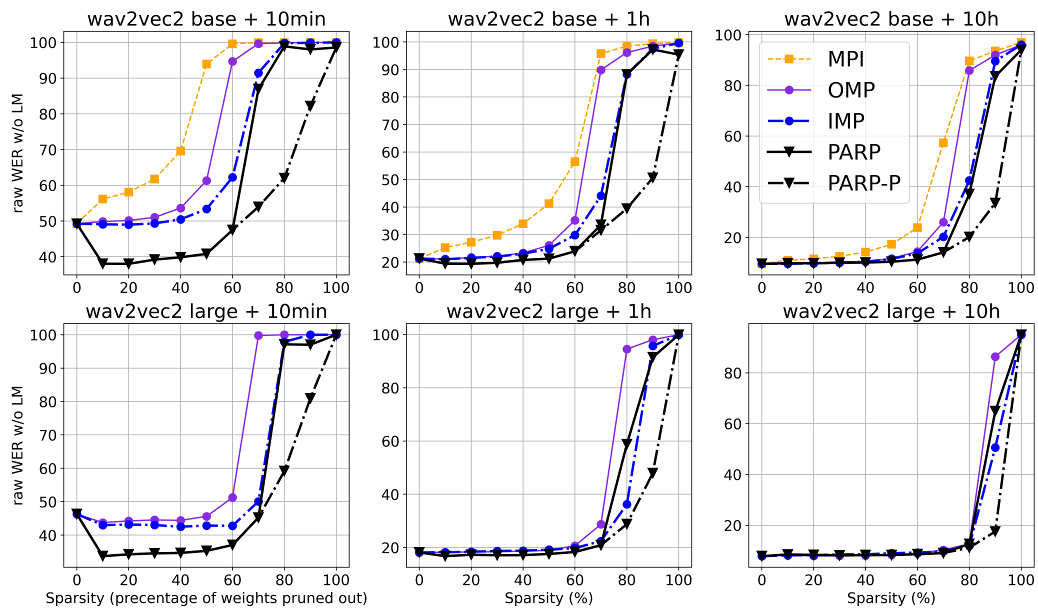


Figure 3-2: Comparison of different pruning techniques on LSR (`wav2vec2` with 10min/1h/10h Librispeech finetuning splits). `PARP` (black line) and `PARP-P` (black dashed line) are especially effective under ultra-low data regime (e.g. 10min) and high-sparsity (70-100%) regions.

Next, we examine if the pruning results of LSR transfers to H2L and CSR. Figure 3-3 is pruning H2L and CSR with 1h of Dutch (*nl*) finetuning, and the same conclusion can be extended to other spoken languages. Comparing Figures 3-2 and 3-3, we notice that shapes of their pruning curves are different, which can be attributed to the effect of character versus phone predictions. Comparing left and center of Figure 3-3, we show that `PARP` and `OMP` reach 50% sparsity on H2L and 70% sparsity on CSR with minimal degradations. Furthermore, while `PARP` is more effective than `OMP` on H2L for all sparsities, such advantage is only visible in the higher sparsity regions on CSR. Lastly, Table 3.3 compares the subnetworks from H2L and CSR with prior work. Even

Table 3.1: WER comparison of pruning LSR: `wav2vec2-base` at 90% sparsity with 10h finetuning on Librispeech without LM decoding. At 90% sparsity, `OMP`/`IMP`/`MPI` perform nearly as bad as `RP`. sub-finetuning stands for subnetwork finetuning.

| Method | # ASR finetunings | test clean | test other |
|---|---|---|---|
| `RP` + sub-finetuning | 1 | 94.5 | 96.4 |
| `MPI` + sub-finetuning | 1 | 93.6 | 96.1 |
| `OMP` + sub-finetuning | 2 | 92.0 | 95.3 |
| `IMP` + sub-finetuning | 10 | 89.6 | 93.9 |
| `PARP` (90% → 90%) | 1 | 83.6 | 90.7 |
| `PARP-P` | | | |
| 70% → 90% | 1 | 51.9 | 69.1 |
| 60% → 80% → 90% | 2 | 33.6 | 53.3 |

Table 3.2: WER comparison of `PARP` for LSR with previous speech SSL results on Librispeech 10min. `PARP` discovers sparse subnetworks within `wav2vec2` with lower WER while adding minimal computational cost to the original ASR finetuning.

| Method | test clean | test other |
|---|---|---|
| Continuous BERT (Baevski et al., 2019a) + LM | 49.5 | 66.3 |
| Discrete BERT (Baevski et al., 2019a) + LM | 16.3 | 25.2 |
| `wav2vec2-base` reported (Baevski et al., 2020) | 46.9 | 50.9 |
| `wav2vec2-large` reported (Baevski et al., 2020) | 43.5 | 45.3 |
| `wav2vec2-base` replicated | 49.3 | 53.2 |
| `wav2vec2-large` replicated | 46.3 | 48.1 |
| `wav2vec2-base` w/ 10% `PARP` | 38.0 | 44.3 |
| `wav2vec2-large` w/ 10% `PARP` | 33.7 | 37.2 |

with as high as 90% sparsities in either settings, subnetworks from `PARP` and `OMP` out-performs prior art.

### 3.4.2 How Important is the Initial Subnetwork (Step 1) in `PARP`?

Obtaining a good initial subnetwork (Step 1) is critical for `PARP`, as Adjust & Re-Prune (Step 2) is operated on top of it. In this section, we isolate the effect of Step 1 from Step 2 and examine the role of the initial subnetwork in `PARP`. Figure 3-4 shows `PARP`

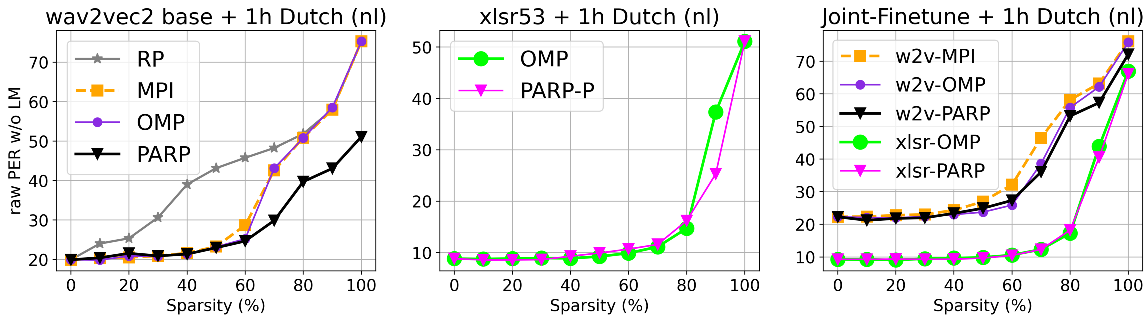


Figure 3-3: Comparison of pruning techniques on H2L & CSR with 1h of Dutch (*nl*) ASR finetuning. **(Left)** Pruning H2L (`wav2vec2-base` + *nl*). **(Center)** Pruning CSR (`xlsr` + *nl*). **(Right)** Pruning jointly-finetuned `wav2vec2-base` and `xlsr` on *nl*. Trend is consistent for other 9 spoken languages.

with a random subnetwork from `RP`, instead of subnetwork from `MPI`, as the initial subnetwork. `PARP` with random initial subnetwork performs nearly as bad as `RP` (grey line), signifying the importance of the initial subnetwork.

Secondly, despite Observation 1, `MPI` in high sparsity regions (e.g. 90% in LSR) is not a good initial subnetwork, since the majority of the weights are already pruned out (thus is hard to be recovered from). From Figure 3-2, `PARP` performs only on par or even worse than `IMP` in high sparsity regions. In contrast, `PARP-P` starts with a relatively lower sparsity (e.g. 60% or 70% `MPI`), and progressively prunes up to the target sparsity. Doing so yields considerable performance gain (up to over 50% absolute WER reduction). Third, as shown in Figure 3.4, there is >99.99% `IOU` between the final "adjusted" subnetwork from `PARP` and its initial `MPI` subnetwork after 20% sparsity, confirming Step 2 indeed only made minimal "adjustment" to the initial subnetwork.

### 3.4.3 Are Pruning Masks Transferrable across Spoken Languages?

Is it possible to discover subnetworks with the wrong guidance, and how transferrable are such subnetworks? More concretely, we investigate the transferability of `OMP` pruning mask discovered from a source language by finetuning its subnetwork on another target language. Such study should shed some insights on the underlying

Table 3.3: Comparing subnetworks discovered by `OMP` and `PARP` from `wav2vec2-base` and `xlsr` with prior work on H2L and CSR. PER is averaged over 10 languages.

| Method | Pre-training | Sparsity | avg. PER |
|---|---|---|---|
| Bottleneck (Fer et al., 2017) | Babel-1070h | 0% | 44.9 |
| CPC (Oord et al., 2018b) | LS-100h | 0% | 50.9 |
| Modified CPC (Rivière et al., 2020) | LS-360h | 0% | 44.5 |
| `wav2vec2-base` | LS-960h | 0% | 18.7 |
| `wav2vec2` + `OMP` | LS-960h | 70% | 41.3 |
| `wav2vec2` + `PARP` | LS-960h | 90% | 40.1 |
| `xlsr` reported (Conneau et al., 2020) | 56,000h | 0% | 7.6 |
| `xlsr` replicated | 56,000h | 0% | 9.9 |
| `xlsr` + `OMP` | 56,000h | 90% | 33.9 |
| `xlsr` + `PARP-P` | 56,000h | 90% | 22.9 |

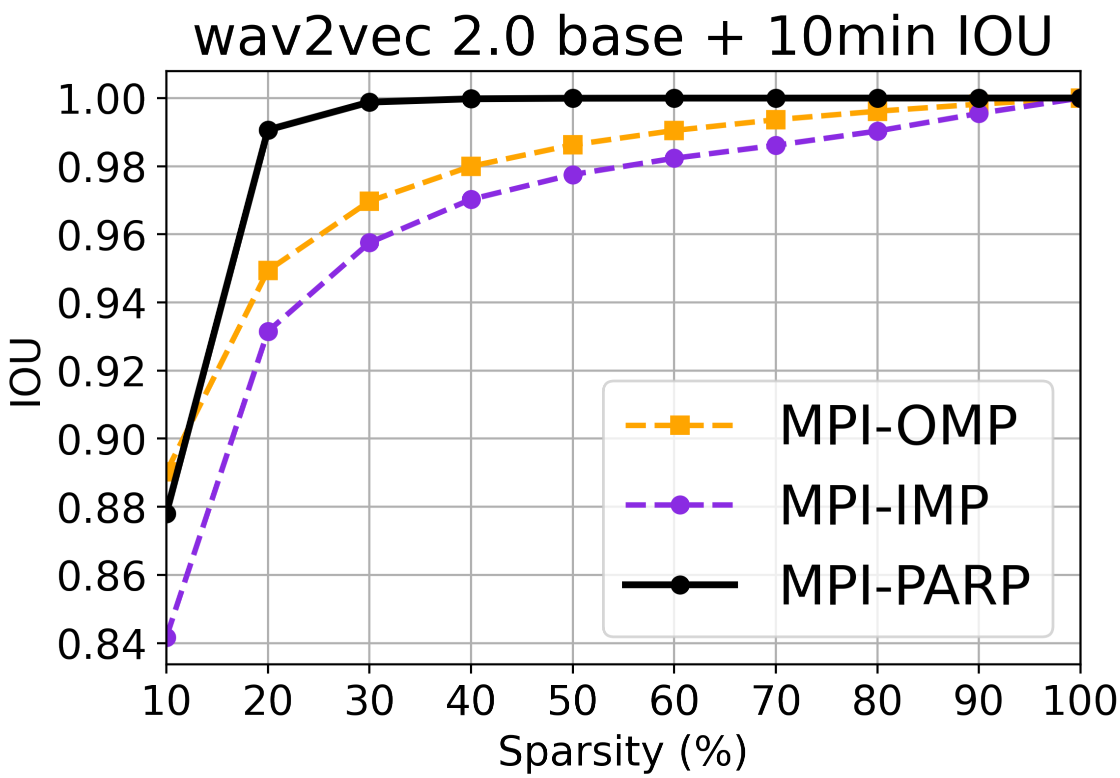


Table 3.4: `PARP`'s final subnetwork and its initial `MPI` subnetwork exceeds 99.99% `IOU` after 20% sparsity (black line).

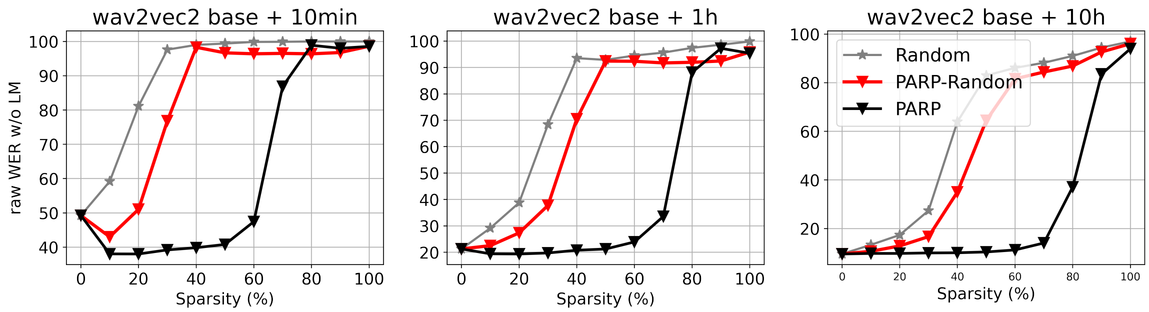


Figure 3-4: `PARP` with random (red line) v.s. with `MPI` (black line) initial subnetworks in LSR.

influence of spoken language structure on network pruning – that similar language pairs should be transferrable. From a practical perspective, consider pruning for an unseen new language in H2L, we could deploy the readily available discovered subnetworks and thus save the additional finetuning and memory costs.

In this case, the initial subnetwork of `PARP` is given by applying `OMP` on another spoken language. According to Observation 1, `PARP`'s Step 2 is effectively under-going cross-lingual subnetwork adaptation for the target language. Figure 3-5 shows the transferability results on H2L with pre-trained `wav2vec2-base`. On the left is a subnetwork at 50% sparsity transfer with regular finetuning that contains subtle language clusters – for example, when finetuning on *ru*, source masks from *es, fr, it, ky, nl* induces a much higher PER compare to that from *sv-SE, tr, tt, zh-TW*. On the right of Figure 3-5, we show that there is no cross-lingual PER degradation with `PARP`, supporting our claim above.

### 3.4.4 Discovering a Single Subnetwork for 10 Spoken Languages

A major downside of pruning pre-trained SSL models for many downstream tasks is the exponential computational and memory costs. In H2L and CSR, the same pruning method needs to be repeatedly re-run for each downstream spoken language at each given sparsity. Therefore, we investigate the possibility of obtaining a single shared subnetwork for all downstream languages. Instead of finetuning separately for each

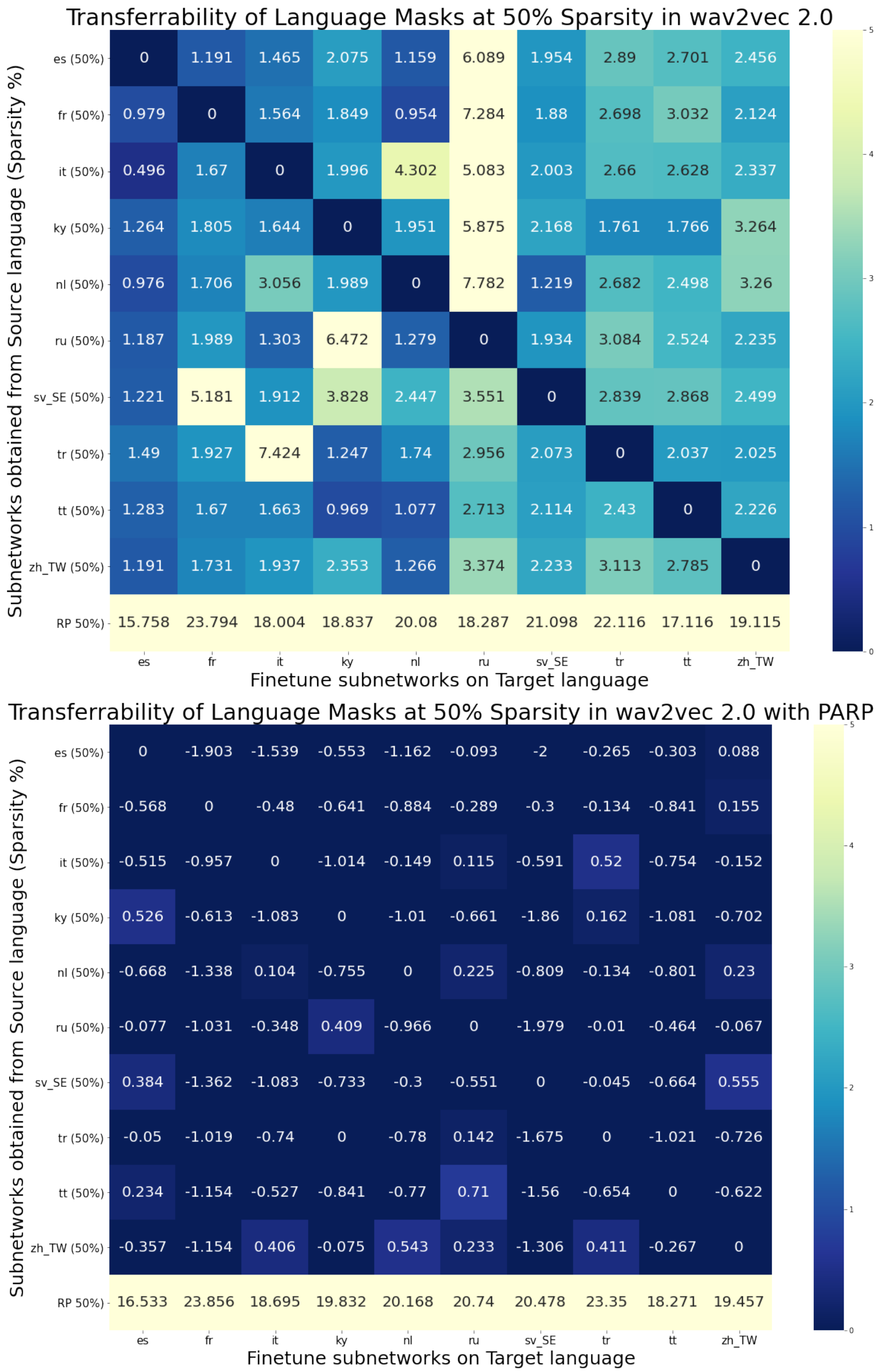


Figure 3-5: (**Left**) Cross-lingual `OMP` mask transfer with regular subnetwork finetuning. (**Right**) Cross-lingual `OMP` mask transfer with PARP. Last rows are `RP`. Values are relative PER gains over same-language pair transfer (hence the darker the bettter). Both are on H2L with pretrained `wav2vec2`.

language, we construct a joint phoneme dictionary and finetune `wav2vec2` and `xlsr` on all 10 languages jointly in H2L and CSR. Note that `PARP` with joint-finetuning can retrieve a shared subnetwork in a single run. The shared subnetwork can then be decoded for each language separately. The right side of Figure 3-3 illustrates the results.

Comparing joint-finetuning and individual-finetuning, in H2L, we found that the shared subnetwork obtained via `OMP` has lower PERs between 60$\sim$80% but slightly higher PERs in other sparsity regions; in CSR, the shared subnetwork from `OMP` has slightly worse PERs at all sparsities. Comparing `PARP` to `OMP` in joint-finetuning, we found that while `PARP` is effective in the individual-finetuning setting (left of Figure 3-3), its shared subnetworks are only slightly better than `OMP` in both H2L and CSR (right of Figure 3-3). The smaller performance gain of `PARP` over `OMP` in pruning jointly-finetuned models is expected, since the important weights for each language are disjoint and joint-finetuning may send mixed signal to the adjustment step in `PARP` (see Figure 3-6 for better illustration).

### 3.4.5 Does `PARP` work on Pre-trained BERT/XLNet?

We also analyzed whether Observation 1 holds for pre-trained BERT/XLNet on 9 GLUE tasks. Surprisingly, we found that there are also high (>98%) overlaps between the 9 tasks' `IMP` pruning masks. Given this observation, we replicated the cross-task subnetwork transfer experiment (take subnetwork found by `IMP` at task A and finetune it for task B) in BERT-Ticket (Chen et al., 2020b) on pre-trained BERT/XLNet with `PARP`. Table 3.5 compares `PARP` (averaged for each target task) to regular finetuning, hinting the applicability of `PARP` to more pre-trained NLP models and downstream natural language tasks.

### 3.4.6 Implications

Observation 1 is consistent with the findings of probing large pre-trained NLP models, that pre-trained SSL models are over-parametrized and there exist task-oriented

Table 3.5: Comparison of cross-task transfer on GLUE (subnetwork from source task A is finetuned for target task B). Numbers are averaged acc. across source tasks for each target task.

| Method | Averaged transferred subnetworks performance finetuned for CoLA | MRPC | QNLI | QQP | RTE | SST-2 | STS-B | WNLI | MNLI |
|---|---|---|---|---|---|---|---|---|---|
| | 70% sparse subnetworks from pre-trained BERT | | | | | | | | |
| Same-task Transfer (top line) | 38.89 | 75.57 | 88.89 | 89.95 | 58.37 | 89.99 | 87.34 | 53.87 | 82.56 |
| Cross-task Transfer with PARP | **28.48** | **75.98** | **87.12** | **90.40** | **59.69** | **89.59** | **86.25** | **54.62** | **81.61** |
| Regular Cross-task Transfer (Chen et al., 2020b) | 10.12 | 71.94 | 86.54 | 88.50 | 57.59 | 88.80 | 80.27 | 54.03 | 80.48 |
| | 70% sparse subnetworks from pre-trained XLNet | | | | | | | | |
| Same-task Transfer (top line) | 29.92 | 76.47 | 89.62 | 90.74 | 59.21 | 92.2 | 80.78 | 42.25 | 85.16 |
| Cross-task Transfer with PARP | **30.09** | **77.56** | **87.10** | **90.66** | **58.88** | **91.73** | **83.80** | **52.11** | **83.87** |
| Regular Cross-task Transfer (Chen et al., 2020b) | 11.47 | 74.16 | 85.21 | 89.11 | 55.80 | 90.19 | 75.61 | 42.25 | 82.65 |

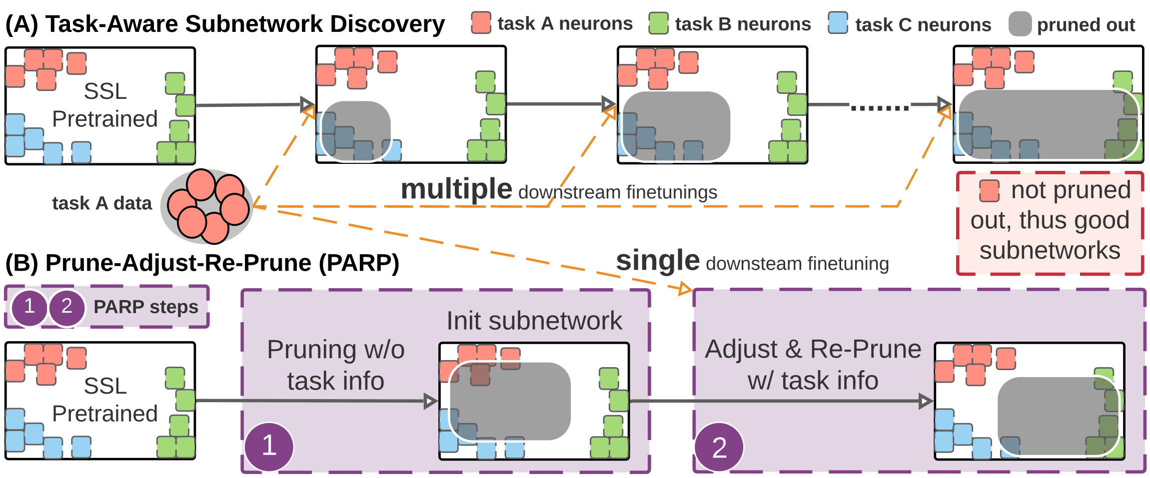


Figure 3-6: Conceptual sketch of pruning the few task-specific important weights in pretrained SSL. **(A)** Task-aware subnetwork discovery(OMP/IMP) is more effective than task-agnostic pruning (MPI) since it foresees the important weights in advance, via multiple downstream finetunings. **(B)** PARP starts with an initial subnetwork given by MPI. Observation 1 suggests that the subnetwork is only off by the few important weights, and thus Step 2 revives them by adjusting the initial subnetwork.

weights/neurons. Figure 3-1 implies that these important weights only account for a small part of the pre-trained speech SSL. In fact, a large body of NLP work is dedicated to studying task-oriented weights in pre-trained models. To name a few, (Durrani et al., 2020; Dalvi et al., 2019; Bau et al., 2018; Xin et al., 2019) measured, (Bau et al., 2018; Dai et al., 2021; Kovaleva et al., 2019) leveraged, (Mu and Andreas, 2020; Goh et al., 2021) visualized, and (Voita et al., 2019; Dalvi et al., 2020; Cao et al., 2021) pruned out these important weights/neurons via probing and quantifying contextualized representations. Based on Observation 1, we can project that these NLP results should in general transfer to speech, see pioneering studies (Belinkov and Glass, 2017; Belinkov et al., 2019; Chung et al., 2021a; Chowdhury et al., 2021). However, different from them, `PARP` leverages important weights for `UMP` on the whole network structure instead of just the contextualized representations.

We could further hypothesize that a good pruning algorithm avoids pruning out task-specific neurons in pre-trained SSL (Lee et al., 2018; Guo et al., 2016; Molchanov et al., 2019), see Figure 3-6. This hypothesis not only offers an explanation on why `PARP` is effective in high sparsity regions and cross-lingual mask transfer, it also suggests that an iterative method such as `IMP` is superior to `OMP` because `IMP` gradually avoids pruning out important weights in several iterations, at the cost of more compute[4]. Finally, we make connections to prior work that showed `RP` prevail (Blalock et al., 2020; Chen et al., 2020b; Liu et al., 2018; Malach et al., 2020; Ramanujan et al., 2020) – under a certain threshold and setting, task-specific neurons are less likely to get "accidentally" pruned and thus accuracy is preserved even with `RP`.

## 3.5 Chapter Summary

In this chapter, we conduct extensive `PARP` and baseline (`OMP` and `IMP`) pruning experiments on low-resource ASR with mono-lingual (pre-trained wav2vec 2.0 (Baevski et al., 2020)) and cross-lingual (pre-trained XLSR-53 (Conneau et al., 2020)) transfer. `PARP` finds significantly superior speech SSL subnetworks for low-resource ASR, while only requiring a single pass of downstream ASR finetuning. Due to its simplicity, `PARP` adds minimal computation overhead to existing SSL downstream finetuning.

- We show that sparse subnetworks exist in pre-trained speech SSL when finetuned for low-resource ASR. In addition, `PARP` achieves superior results to `OMP` and `IMP` across all sparsities, amount of finetuning supervision, pre-trained model scale,

[4]From Section 6 of (Frankle and Carbin, 2018): "iterative pruning is computationally intensive, requiring training a network 15 or more times consecutively for multiple trials." From Section 1 of (Guo et al., 2016): "several iterations of alternate pruning and retraining are necessary to get a fair compression rate on AlexNet, while each retraining process consists of millions of iterations, which can be very time consuming."

and downstream spoken languages. Specifically, on Librispeech 10min without LM decoding, `PARP` discovers subnetworks from wav2vec 2.0 with an absolute 10.9%/12.6% WER decrease compared to the full model, without modifying the finetuning hyper-parameters or objective (Section 3.4.1).

- `PARP` minimizes phone recognition error increases in cross-lingual mask transfer, where a subnetwork pruned for ASR in one spoken language is adapted for ASR in another language (Section 3.4.3). `PARP` can also be applied to efficient multi-lingual subnetwork discovery for 10 spoken languages (Section 3.4.4).

- Last but not least, we demonstrate `PARP`'s effectiveness on pre-trained BERT/XL-Net, mitigating the cross-task performance degradation reported in BERT-Ticket (Chen et al., 2020b) (Section 3.4.5).

# Chapter 4

# Finding Sparse Subnetworks in End-to-End Speech Synthesis

## 4.1 Introduction

End-to-end text-to-speech (TTS)[1] research has focused heavily on modeling techniques and architectures, aiming to produce more natural, adaptive, and expressive speech in robust, low-resource, controllable, or online conditions (Tan et al., 2021). We argue that an overlooked orthogonal research direction in end-to-end TTS is *architectural efficiency*, and in particular, there has not been any established study on pruning end-to-end TTS in a principled manner. As the body of TTS research moves toward the mature end of the spectrum, we expect a myriad of effort delving into developing efficient TTS, with direct implications such as on-device TTS or a better rudimentary understanding of training TTS models from scratch (Frankle and Carbin, 2018).

To this end, this chapter covers analyses on the effects of pruning end-to-end TTS, utilizing basic unstructured magnitude-based weight pruning[2]. The overarching message we aim to deliver is two-fold:

[1]We refer to end-to-end TTS systems as those composed of an acoustic model (also known as text-to-spectrogram prediction network) and a separate vocoder, as there are relatively few direct text-to-waveform models; see (Tan et al., 2021).

[2]Given that there has not been a dedicated TTS pruning study in the past, we resort to the most basic form of pruning. For more advanced pruning techniques, please refer to (Gale et al., 2019; Blalock et al., 2020).

- End-to-end TTS models are over-parameterized; their weights can be pruned with unstructured magnitude-based methods.

- Pruned models can produce synthetic speech at equal or even better naturalness and intelligibility with similar prosody.

### 4.1.1 Background

To introduce our work, we first review two areas of related work:

**Efficiency in TTS** One line of work is on small-footpoint, fast, and parallelizable versions of WaveNet (Oord et al., 2016) and WaveGlow (Prenger et al., 2019) vocoders; prominent examples are WaveRNN[3] (Kalchbrenner et al., 2018), WaveFlow (Ping et al., 2020), Clarinet (Ping et al., 2019), HiFi-GAN (Kong et al., 2020a), Parallel WaveNet (Oord et al., 2018a), SqueezeWave (Zhai et al., 2020), DiffWave (Kong et al., 2021), WaveGrad 1 (Chen et al., 2021a), Parallel WaveGAN (Yamamoto et al., 2020) etc. Another is acoustic models based on non-autoregressive generation (ParaNet (Peng et al., 2020), Flow-TTS (Miao et al., 2020), MelGAN (Kumar et al., 2019), EfficientTTS (Miao et al., 2021), FastSpeech (Ren et al., 2019, 2021)), neural architecture search (LightSpeech (Luo et al., 2021)), diffusion (WaveGrad 2 (Chen et al., 2021b)), etc. Noticeably, efficient music generation has gathered attention too, e.g. NEWT (Hayes et al., 2021) and DDSP (Engel et al., 2020).

**ASR Pruning** Earlier work on ASR pruning reduces the FST search space, such as (Xu et al., 2018). More recently, the focus has shifted to pruning end-to-end ASR models (Yu et al., 2012; Shangguan et al., 2019; Wu et al., 2021; Lai et al., 2021b). Generally speaking, pruning techniques proposed for vision models (Gale et al., 2019; Blalock et al., 2020) work decently well in prior ASR pruning work, which leads us to ask, how effective are simple pruning techniques for TTS?

[3]Structured pruning was in fact employed in WaveRNN, but merely for reducing memory overhead for the vocoder. What sets this work apart is our pursuit of the scientific aspects of pruning end-to-end TTS holistically.

## 4.2 Preliminaries

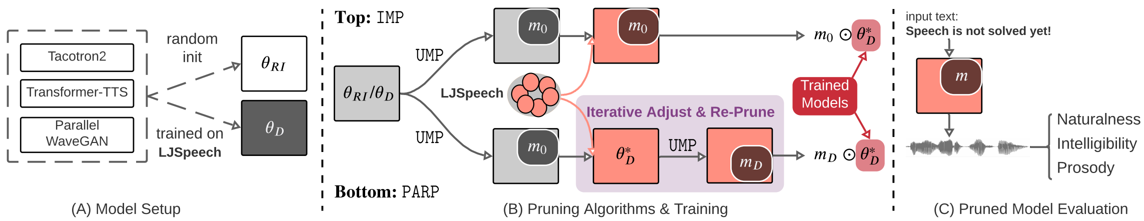


Figure 4-1: Illustration of our end-to-end TTS pruning setup. **Left:** three TTS models are considered: Tacotron2, Transformer-TTS, and Parallel WaveGAN. By default, we set the initial weights $\theta_0$ to trained models $\theta_D$ on LJSpeech, but they can also be randomly initialized $\theta_{RI}$. **Middle:** top row is the `IMP` Baseline, and bottom row is `PARP`. Both are architecture-agnostic, and utilize `UMP` for retrieving initial pruning mask $m_0$. The only difference is that $m_0$ is adjustable in `PARP` during training, while being fixed in `IMP`. Both algorithms produce pruned subnetworks $m \odot \theta_D^*$ that are finetuned on LJSpeech. **Right:** we evaluate pruned model synthetic speech's naturalness, intelligibility, and prosody via large-scale subjective and objective tests across sparsities.

**Prune-Adjust-Re-Prune (`PARP`)** (Lai et al., 2021b) is a simple modified version of `IMP` recently proposed for self-supervised speech recognition, showing that pruned wav2vec 2.0 (Baevski et al., 2020) attains lower WERs than the full model under low-resource conditions. Given its simplicity, here we show that `PARP` can be applied to any sequence-to-sequence learning scenario. Similarly, given an initial model weight $\theta_0$ and $\mathcal{D}$, `PARP` can be described as (See Fig 4-1 for visualization):

1. Same as `IMP`'s Step 1.

2. Train $f(\boldsymbol{X}; \theta_0)$ on $\mathcal{D}$. Zeroed-out weights in $\theta_0$ receive gradient updates via backprop. After $N$ model updates, obtain the trained model $f(\boldsymbol{X}; \theta_D^*)$, and apply `UMP` on $\theta_D^*$ to obtain mask $m_D$. Return subnetwork $m_D \odot \theta_D^*$.

**Setting Initial Model Weight $\theta_0$** In (Lai et al., 2021b), `PARP`'s $\theta_0$ can be the self-supervised pretrained initializations, or any trained model weight $\theta_P$ ($P$ needs not be the target task $D$). On the other hand, `IMP`'s $\theta_0$ is target-task dependent i.e. $\theta_0$ is set to a trained weight on $\mathcal{D}$, denoted as $\theta_D$. However, since the focus in this work is on the final pruning performance only, we set $\theta_0$ to $\theta_D$ by default for both `PARP` and `IMP`.

**Progressive Pruning with `PARP-P`** Following (Lai et al., 2021b), we also experiment with progressive pruning (`PARP-P`), where `PARP-P`'s Step 1 prunes $\theta_0$ at a lower sparsity, and its Step 2 progressively prunes to the target sparsity every $N$ model updates. We show later that `PARP-P` is especially effective in higher sparsity regions.

## 4.3 Experimental Setup

### 4.3.1 TTS Models and Data

**Model Configs** Our end-to-end TTS is based on an acoustic model (phone to melspec) and a vocoder (melspec to wav). To ensure reproducibility, we used publicly available and widely adopted implementations[4]: Transformer-TTS (Li et al., 2019) and Tacotron2 (Shen et al., 2018) as the acoustic models, and Parallel WaveGAN (Yamamoto et al., 2020) as the vocoder. Transformer-TTS and Tacotron2 have the same high-level structure (encoder, decoder, pre-net, post-net) and loss (l2 reconstructions before and after post-nets and stop token cross-entropy). Transformer-TTS consists of a 6-layer encoder and a 6-layer decoder. Tacotron2's encoder consists of 3-layer convolutions and a BLSTM, and its decoder is a 2-layer LSTM with attention. Both use a standard G2P for converting text to phone sequences as the model input. Parallel WaveGAN consists of convolution-based generator $G$ and discriminator $D$.

**Datasets** LJspeech (Ito and Johnson, 2017) is used for training acoustic models and vocoders. It is a female single-speaker read speech corpus with 13k text-audio pairs, totaling 24h of recordings. We also used the transcription of Librispeech's `train-clean-100` partition (Panayotov et al., 2015) as additional unspoken text[5] used in TTS-Augmentation.

[4]Checkpoints are also available at ESPnet and ParallelWaveGAN.

[5]Both LJspeech and Librispeech are based on audiobooks.

### 4.3.2 `PARP` Implementation

`UMP` is based on PyTorch's API[6]. For all models, $\theta_0$ is set to pretrained checkpoints on LJspeech, and $N$ is set to 1 epoch of model updates. We jointly prune encoder, decoder, pre-nets, and post-nets for the acoustic model; for vocoder, since only $G$ is needed during test-time synthesis, only $G$ is pruned ($D$ is still trainable).

### 4.3.3 Complementary Techniques for `PARP`

**TTS-Augmentation for unspoken transcriptions** The first technique is based on TTS-Augmentation (Hwang et al., 2021). It is a form of self-training, where we take $f(\theta_D)$ to label additional unspoken text $\boldsymbol{X}_u$. The newly synthesized paired data, denoted $\mathcal{D}_u = (\boldsymbol{X}_u, f(\boldsymbol{X}_u; \theta_D))$, is used together with $\mathcal{D}$ in `PARP`'s Step 2.

**Combining Knowledge-Distillation (`KD`) and `PARP`,** with a teacher model denoted as $f(\theta_D)$. The training objective in `PARP`'s Step 2 is set to reconstructing both ground truth melspec and melspec synthesized by an (unpruned) teacher acoustic model $f(\theta_D)$.

### 4.3.4 Subjective and Objective Evaluations

We examine the following three aspects of the synthetic speech:

- **Naturalness** is quantified by the 5-point (1-point increment) scale Mean Opinion Score (MOS). 20 unique utterances (with 5 repetitions) are synthesized and compared across pruned models, for a total of 100 HITs (crowdsourced tasks) per MOS test. In each HIT, the input texts to all models are the same to minimize variability.

- **Intelligibility** is measured with Google's ASR API[7].

- **Prosody** via mean and standard deviation (std) fundamental frequency ($F_0$) estimations[8] and utterance duration, averaged over dev and eval utterances.

[6]PyTorch Pruning API
[7]`https://pypi.org/project/SpeechRecognition/`
[8]$F_0$ estimation with probabilistic YIN (pYIN) implemented in Librosa.

We also perform pairwise comparison (A/B) testings for naturalness and intelligibility (separately). Similar to our MOS test, we release 20 unique utterances (with 10 repetitions), for a total of 200 HITs per A/B test. In each HIT, input text to models are also the same. MOS and A/B tests are conducted in Amazon Mechanical Turk (AMT).

**Statistical Testing** To ensure our AMT results are statistically significant, we run Mann-Whitney U test for each MOS test, and pairwise z-test for each A/B test, both at significance level of $p \leq 0.05$.

## 4.4 Results

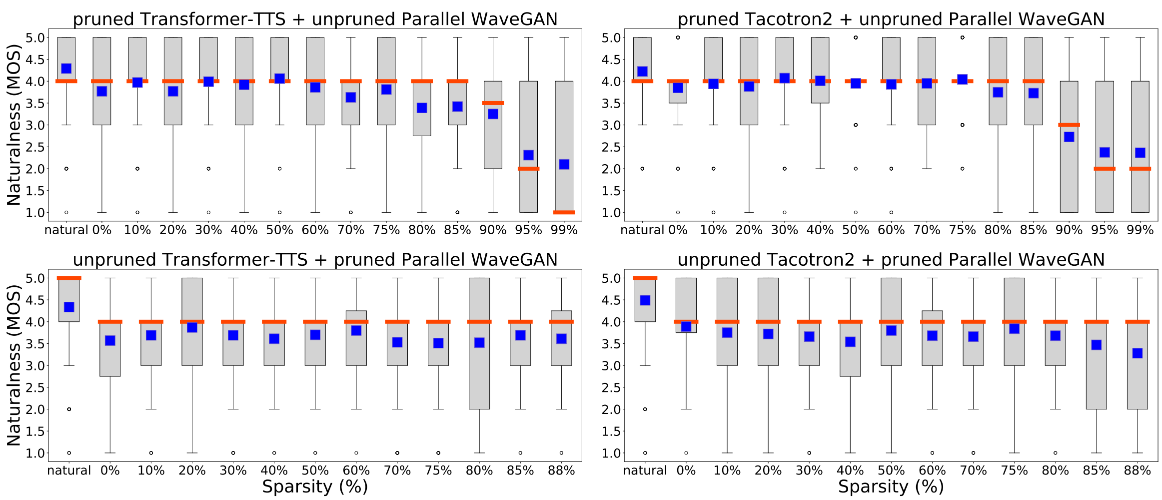


Figure 4-2: Box plots for four independent MOS tests across configurations (pruned/unpruned acoustic models + pruned/unpruned vocoders). At each sparstiy, ■ is the mean and ▬ is the median MOS score over 100 HITs. Ground truth recordings (natural) are included as the topline.

### 4.4.1 Does Sparsity improve Naturalness?

Fig 4-2 is the box plot of MOS scores of pruned end-to-end TTS models at 0%∼99% sparsities with `PARP`. In each set of experiments, only one of the acoustic model or vocoder is pruned, while the other is kept intact. For either pruned Transformer-TTS or Tacotron2 acoustic models, their MOS scores are statistically not different from the unpruned ones at up to 90% sparsity. For pruned Parallel WaveGAN, pairing it with an unpruned Transformer-TTS reaches up to 88% sparsity without *any* statistical MOS decrease, and up to 85% if paired with an unpruned Tacotron2. Based on these results, we first conclude that end-to-end TTS models are over-parameterized across model architectures, and removing the majority of their weights does not significantly affect naturalness.

Secondly, we observe that the 30% pruned Tacotron2 has a statistically higher MOS score than unpruned Tacotron2. Although this phenomenon is not seen in Transformer-TTS, WaveGAN, or at other sparsities, it is nonetheless surprising given `PARP`'s simplicity. We can hypothesize that under the right conditions, *pruned models train better*, which results in higher naturalness over unpruned models.

### 4.4.2 Does Sparsity improve Intelligibility?

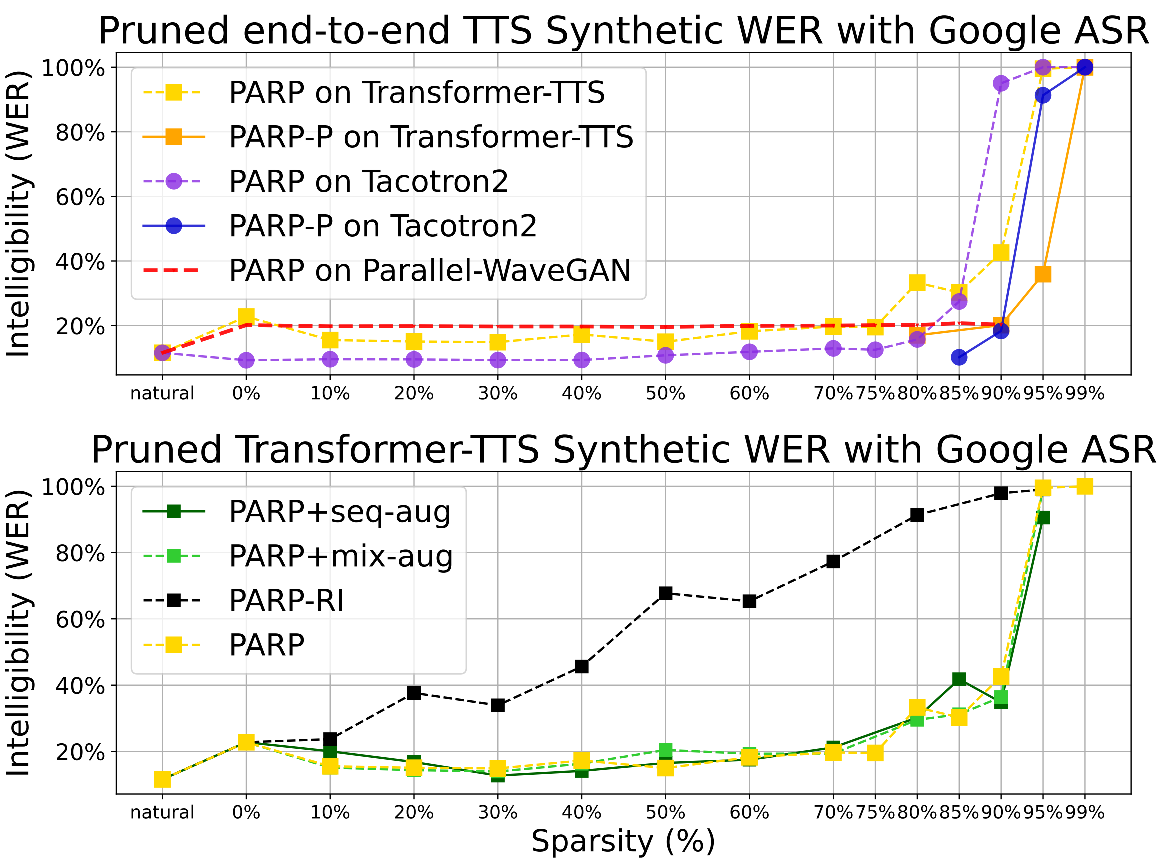


Figure 4-3: **Top** plots the synthetic speech WERs over sparsities for all model combinations. **Bottom** compares the WERs for different pruning configurations.

We measure intelligibility of synthetic speech via Google ASR, and Figure 4-3 plots synthetic speech's WERs across sparsities over model and pruning configurations. Focusing on the top plot, we have the following two high-level impressions: (1) WER decreases at initial sparsities and increases dramatically at around 85% sparsity with `PARP` (yellow and purple dotted lines). (2) pruning the vocoder does not change the WERs at all (observe the straight red dotted line).

Specifically, for Transformer-TTS, `PARP` at 75% and `PARP-P` at 90% sparsities have lower WERs (higher intelligibility) than its unpruned version. For Tacotron2, there is no WER reduction and its WERs remain at ∼9% at up to 40% sparsity (no change in intelligibility). Based on (2) and Section 4.4.1, we can further conclude that the CNN-based vocoder is highly prunable, with little to no naturalness and intelligibility degradation at up to almost 90% sparsity.

### 4.4.3 Does Sparsity change Prosody?

We used synthetic speech's utterance duration and mean/std $F_0$ across time as three rough proxies for prosody. Fig 4-4 plots the prosody mismatch between pruned models and ground truth recordings across model combinations. Observe `PARP` on Tacotron2 and on Transformer-TTS result in visible differences in prosody changes over sparsities. In the top plot, pruned Transformer-TTS (yellow dotted line) have the same utterance duration (+0.2 seconds over ground truth) at 10%∼75% sparsities, while in the same region, pruned Tacotron2 (purple dotted line) results in a linear decrease in duration (-0.2∼-0.8 seconds). Indeed, we confirmed by listening to synthesis samples that pruning Tacotron2 leads to shorter utterance duration as sparsity increases.

In the middle plot and up to 80% sparsity, pruned Tacotron2 models have a much large $F_0$ mean variation (-20∼-7.5 Hz) compared to that of Transformer-TTS (-10∼-15 Hz). We hypothesize that `PARP` on RNN-based models leads to unstable gradients through time during training, while Transformer-based models are easier to prune. Further, `PARP` on WaveGAN (red dotted line) has a minimal effect on both metrics across sparsities, which leads us to another hypothesis that *vocoder is not responsible for prosody generation.*

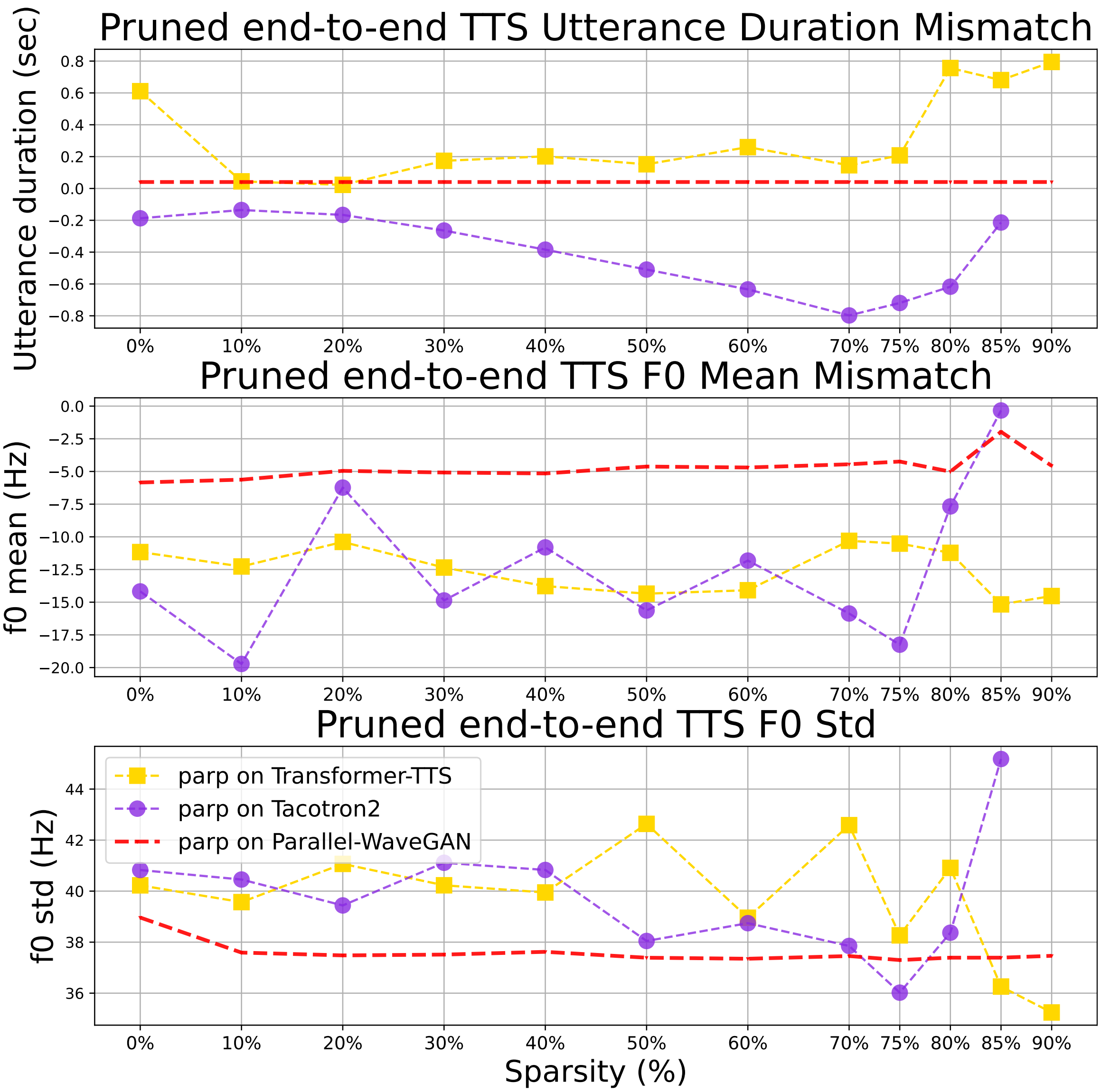


Figure 4-4: **Top** is utterance duration mismatch (in seconds), **Middle** is $F_0$ mean mismatch (in Hz), and **Bottom** is $F_0$ std (in Hz). Mismatches are calculated against ground truth recordings. Full model (0%) results are also included.

In the bottom plot and up to 80% sparsity, pruned models all have minimal $F_0$ std variations ($\leq$ 2 Hz) compared to 53Hz ground truth $F_0$ std. We infer that at reasonable sparsities, *pruning does not hurt prosodic expressivity*, due to lack of $F_0$ oversmoothing (Tan et al., 2021; Zen et al., 2009).

### 4.4.4 Does more finetuning data improve sparsity?

In (Lai et al., 2021b), the authors attain pruned wav2vec 2.0 at much higher sparsity without WER increase given sufficient finetuning data (10h Librispeech split). Therefore, one question we had was, how much finetuning data is “good enough" for pruning end-to-end TTS? We did two sets of experiments, and for each, we modify the amount of data in PARP's Step 2, while keeping $\theta_0$ as is (trained on full LJspeech).

The first set of experiments result is Fig 4-5. Even at as high as 90% sparsity, 30% of finetuning data (∼7.2h) is enough for PARP to reach the same level of naturalness as full data[9]. The other set of experiment is TTS-Augmentation for utilizing additional unspoken text (∼100h, no domain mismatch) for PARP's Step 2. In Fig 4-3's bottom plot, we see TTS-Augmentations (dark & light green lines) bear minimal effect on the synthetic speech WERs. However, Table 4.1 indicates that TTS-Augmentation PARP+seq-aug does statistically improve PARP in naturalness and intelligibility subjective testings.

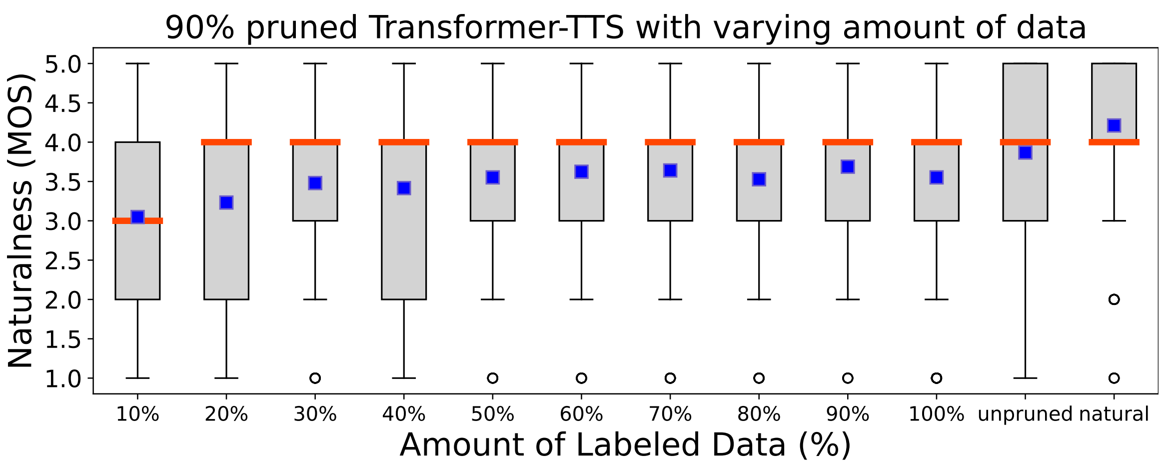


Figure 4-5: Effect of amount of finetuning data in PARP's Step 2 on MOS score. Model is 90% pruned Transformer-TTS.

### 4.4.5 Ablations

**Knowledge Distillation hurts PARP** Surprisingly, we found combining knowledge distillation from teacher model $f(\theta_D)$ with PARP significantly reduces the synthesis

[9]The effect of using less data to obtain $\theta_0$ remains unclear.

quality, see PARP+KD v.s. PARP in Table 4.1. Perhaps more careful tuning is required to make KD work.

**Importance of $\theta_0$** Bottom plot of Fig 4-3 (black dotted line) and Table 4.1 (PARP v.s. PARP-RI) demonstrate the importance of setting the initial model weight $\theta_0$. In both cases, we set $\theta_0$ to random initialization (RI) instead of $\theta_D$ on LJspeech.

**Effectiveness of IMP** Table 4.1 shows the clear advantage of PARP-P over IMP at high sparsities, yet PARP is not strictly better than IMP.

Table 4.1: A/B testing results. Each comparison is over 200 HITs. **Bold** numbers are statistical significant under pairwise z test.

| Proposal | Baseline | Sparsity | Preference over Baseline | |
|---|---|---|---|---|
| | | Level | Naturalness | Intelligibility |
| **pruned Transformer-TTS + unpruned Parallel WaveGAN** | | | | |
| PARP-P | PARP | 90% | **57%** | **66%** |
| | | 95% | **63%** | **64%** |
| PARP+KD | PARP | 70% | **40%** | **43%** |
| | | 90% | **36%** | **27%** |
| PARP-P | IMP | 90% | 53% | 51% |
| | | 95% | **64%** | **61%** |
| PARP | IMP | 30% | 54% | **58%** |
| | | 50% | 46% | 54% |
| | | 90% | **42%** | **37%** |
| PARP | PARP-RI | 10% | 55% | **57%** |
| | | 30% | 55% | 53% |
| | | 50% | **56%** | **67%** |
| | | 70% | 53% | 53% |
| | | 90% | **60%** | **56%** |
| PARP+seq-aug | PARP | 10% | **58%** | **58%** |
| | | 30% | 52% | **57%** |
| | | 50% | **44%** | **41%** |
| | | 70% | **57%** | 54% |
| | | 90% | 51% | **56%** |

## 4.5 Chapter Summary

This chapter builds upon a recent ASR pruning technique termed PARP (Lai et al., 2021b), with the intention of not only reducing architectural complexity for end-to-end

TTS, but also demonstrating the surprising efficacy and simplicity of pruning in contrast to prior TTS efficiency work. Our contributions are:

- We present the first comprehensive study on pruning end-to-end acoustic models (Transformer-TTS (Li et al., 2019), Tacotron2 (Shen et al., 2018)) and vocoders (Parallel WaveGAN (Yamamoto et al., 2020)) with an unstructured magnitude based pruning method `PARP` (Lai et al., 2021b).

- We extend `PARP` with knowledge distillation (`KD`) and TTS-Augmentation (Hwang et al., 2021) for TTS pruning, demonstrating `PARP`'s applicability and effectiveness regardless of network architectures or input/output pairs.

- We show that end-to-end TTS models are over-parameterized. Pruned models produce speech with similar levels of naturalness, intelligibility, and prosody to that of unpruned models.

- For instance, with large-scale subjective tests and objective measures, Tacotron2 at 30% sparsity has statistically better naturalness than its original version; for another, small footprint CNN-based vocoder has little to no synthesis degradation at up to 88% sparsity.

# Chapter 5

# Conclusion

This thesis proposes a simple and intuitive pruning method, `PARP`, for self-supervised speech recognition and end-to-end speech synthesis. On the high-level, we show that sparse subnetworks exist in modern speech processing models, and sparse subnetworks attain similar performance as dense networks in recognition and synthesis.

**Summary of Results.** In the first study, we conduct experiments on pruning pre-trained wav2vec 2.0 and XLSR-53 under three low-resource settings, demonstrating (1) `PARP` discovers better subnetworks than baseline pruning methods while requiring a fraction of their computational cost, (2) the discovered subnetworks yields over 10% WER reduction over the full model, (3) `PARP` induces minimal cross-lingual subnetwork adaptation errors, (4) `PARP` can discover a shared subnetwork for multiple spoken languages in one pass, and (5) `PARP` significantly reduces cross-task adaptation errors of pre-trained BERT/XLNet. In the second study, we then demonstrate `PARP`'s effectiveness by pruning transformer-TTS and Parallel WaveGAN, finding the pruned TTS models produce synthetic speech at equal or even better naturalness and intelligibility with similar prosody. Beyond the scope of our study, we aspire `PARP` as the beginning of many future endeavours on developing more efficient speech processing models.

**Broader Impact.** The broader impact of this thesis is making speech technologies more accessible in two orthogonal dimensions: (i) extending modern-day speech technology to many under-explored low-resource spoken languages, and (ii) introducing a new and flexible pruning technique to current and future speech processing models that reduces the computational costs required for adapting (finetuning) them to custom settings.

# Bibliography


Sherif Abdou and Michael S Scordilis. 2004. Beam search pruning in speech recognition using a posterior probability-based confidence measure. *Speech Communication*, 42(3-4):409–428.

Junyi Ao, Rui Wang, Long Zhou, Shujie Liu, Shuo Ren, Yu Wu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, et al. 2021. Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing. *arXiv preprint arXiv:2110.07205*.

Alexei Baevski, Michael Auli, and Abdelrahman Mohamed. 2019a. Effectiveness of self-supervised pre-training for speech recognition. *arXiv preprint arXiv:1911.03912*.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2021. Unsupervised speech recognition. *arXiv preprint arXiv:2105.11084*.

Alexei Baevski, Steffen Schneider, and Michael Auli. 2019b. vq-wav2vec: Self-supervised learning of discrete speech representations. *arXiv preprint arXiv:1910.05453*.

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*.

Anthony Bau, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. 2018. Identifying and controlling important neurons in neural machine translation. *arXiv preprint arXiv:1811.01157*.

Yonatan Belinkov, Ahmed Ali, and James Glass. 2019. Analyzing phonetic and graphemic representations in end-to-end automatic speech recognition. *arXiv preprint arXiv:1907.04224*.

Yonatan Belinkov and James Glass. 2017. Analyzing hidden representations in end-to-end automatic speech recognition systems. *arXiv preprint arXiv:1709.04482*.

Guillaume Bellec, David Kappel, Wolfgang Maass, and Robert Legenstein. 2017. Deep rewiring: Training very sparse deep networks. *arXiv preprint arXiv:1711.05136*.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*.

Stefan Braun and Shih-Chii Liu. 2019. Parameter uncertainty for end-to-end speech recognition. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5636–5640. IEEE.

Steven Cao, Victor Sanh, and Alexander M Rush. 2021. Low-complexity probing via finding subnetworks. *arXiv preprint arXiv:2104.03514*.

William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi. 2021. Speechstew: Simply mix all available speech recognition data to train one large neural network. *arXiv preprint arXiv:2104.02133*.

Xuankai Chang, Takashi Maekaku, Pengcheng Guo, Jing Shi, Yen-Ju Lu, Aswin Shanmugam Subramanian, Tianzi Wang, Shu-wen Yang, Yu Tsao, Hung-yi Lee, et al. 2021. An exploration of self-supervised pretrained representations for end-to-end speech recognition. *arXiv preprint arXiv:2110.04590*.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. 2021a. Wavegrad: Estimating gradients for waveform generation. *ICLR*.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, Najim Dehak, and William Chan. 2021b. Wavegrad 2: Iterative refinement for text-to-speech synthesis. *Interspeech*.

Sanyuan Chen, Yu Wu, Zhuo Chen, Jian Wu, Jinyu Li, Takuya Yoshioka, Chengyi Wang, Shujie Liu, and Ming Zhou. 2021c. Continuous speech separation with conformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5749–5753. IEEE.

Sanyuan Chen, Yu Wu, Chengyi Wang, Zhengyang Chen, Zhuo Chen, Shujie Liu, Jian Wu, Yao Qian, Furu Wei, Jinyu Li, et al. 2021d. Unispeech-sat: Universal speech representation learning with speaker aware pre-training. *arXiv preprint arXiv:2110.05752*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Michael Carbin, and Zhangyang Wang. 2020a. The lottery tickets hypothesis for supervised and self-supervised pre-training in computer vision models. *arXiv preprint arXiv:2012.06908*.

Tianlong Chen, Jonathan Frankle, Shiyu Chang, Sijia Liu, Yang Zhang, Zhangyang Wang, and Michael Carbin. 2020b. The lottery ticket hypothesis for pre-trained bert networks. *arXiv preprint arXiv:2007.12223*.

Yi-Chen Chen, Shu-wen Yang, Cheng-Kuang Lee, Simon See, and Hung-yi Lee. 2021e. Speech representation learning through self-supervised pretraining and multi-task finetuning. *arXiv preprint arXiv:2110.09930*.

Jaejin Cho, Murali Karthick Baskar, Ruizhi Li, Matthew Wiesner, Sri Harish Mallidi, Nelson Yalta, Martin Karafiat, Shinji Watanabe, and Takaaki Hori. 2018. Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 521–527. IEEE.

Jan Chorowski, Grzegorz Ciesielski, Jarosław Dzikowski, Adrian Łancucki, Ricard Marxer, Mateusz Opala, Piotr Pusz, Paweł Rychlikowski, and Michał Stypułkowski. 2021. Aligned contrastive predictive coding. *arXiv preprint arXiv:2104.11946*.

Shammur Absar Chowdhury, Nadir Durrani, and Ahmed Ali. 2021. What do end-to-end speech models learn about speaker, language and channel information? a layer-wise and neuron-level analysis. *arXiv preprint arXiv:2107.00439*.

Yu-An Chung, Yonatan Belinkov, and James Glass. 2021a. Similarity analysis of self-supervised speech representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3040–3044. IEEE.

Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass. 2019. An unsupervised autoregressive model for speech representation learning. *arXiv preprint arXiv:1904.03240*.

Yu-An Chung, Wei-Hung Weng, Schrasing Tong, and James Glass. 2018. Unsupervised cross-modal alignment of speech and text embedding spaces. *arXiv preprint arXiv:1805.07467*.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021b. W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. *arXiv preprint arXiv:2108.06209*.

Yu-An Chung, Chenguang Zhu, and Michael Zeng. 2020. Splat: Speech-language joint pre-training for spoken language understanding. *arXiv preprint arXiv:2010.02295*.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2020. Unsupervised cross-lingual representation learning for speech recognition. *arXiv preprint arXiv:2006.13979*.

Erica Cooper, Wen-Chin Huang, Tomoki Toda, and Junichi Yamagishi. 2021. Generalization ability of mos prediction networks. *arXiv preprint arXiv:2110.02635*.

Jia Cui, Xiaodong Cui, Bhuvana Ramabhadran, Janice Kim, Brian Kingsbury, Jonathan Mamou, Lidia Mangu, Michael Picheny, Tara N Sainath, and Abhinav Sethy. 2013. Developing speech recognition systems for corpus indexing under the iarpa babel program. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6753–6757. IEEE.

Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislal, Lidia Mangu, Markus Nussbaum-Thom, Michael Picheny, et al. 2015. Multilingual representations for low resource speech recognition and keyword search. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 259–266. IEEE.

Xiaodong Cui, Brian Kingsbury, Jia Cui, Bhuvana Ramabhadran, Andrew Rosenberg, Mohammad Sadegh Rasooli, Owen Rambow, Nizar Habash, and Vaibhava Goel. 2014. Improving deep neural network acoustic modeling for audio corpus indexing under the iarpa babel program. In *Fifteenth Annual Conference of the International Speech Communication Association*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv preprint arXiv:2104.08696*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6309–6317.

Fahim Dalvi, Hassan Sajjad, Nadir Durrani, and Yonatan Belinkov. 2020. Analyzing redundancy in pretrained transformer models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4908–4926.

Nadir Durrani, Hassan Sajjad, Fahim Dalvi, and Yonatan Belinkov. 2020. Analyzing individual neurons in pre-trained language models. *arXiv preprint arXiv:2010.02695*.

Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. 2020. Ddsp: Differentiable digital signal processing. *ICLR*.

Radek Fer, Pavel Matějka, František Grézl, Oldřich Plchot, Karel Veselỳ, and Jan Honza Černockỳ. 2017. Multilingually trained bottleneck features in spoken language recognition. *Computer Speech & Language*, 46:252–267.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Jonathan Frankle, Gintare Karolina Dziugaite, Daniel Roy, and Michael Carbin. 2020. Linear mode connectivity and the lottery ticket hypothesis. In *International Conference on Machine Learning*, pages 3259–3269. PMLR.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*.

Mark JF Gales, Kate M Knill, Anton Ragni, and Shakti P Rath. 2014. Speech recognition and keyword spotting for low-resource languages: Babel project research at cued. In *Fourth International Workshop on Spoken Language Technologies for Under-Resourced Languages (SLTU-2014)*, pages 16–23. International Speech Communication Association (ISCA).

Dawei Gao, Xiaoxi He, Zimu Zhou, Yongxin Tong, Ke Xu, and Lothar Thiele. 2020. Rethinking pruning for accelerating deep inference at the edge. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 155–164.

Heting Gao, Junrui Ni, Yang Zhang, Kaizhi Qian, Shiyu Chang, and Mark Hasegawa-Johnson. 2021. Zero-shot cross-lingual phonetic recognition with external language embedding. *Proc. Interspeech 2021*, pages 1304–1308.

Sharath Girish, Shishira R Maiya, Kamal Gupta, Hao Chen, Larry Davis, and Abhinav Shrivastava. 2020. The lottery ticket hypothesis for object recognition. *arXiv preprint arXiv:2012.04643*.

Gabriel Goh, Nick Cammarata, Chelsea Voss, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. 2021. Multimodal neurons in artificial neural networks. *Distill*, 6(3):e30.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Yiwen Guo, Anbang Yao, and Yurong Chen. 2016. Dynamic network surgery for efficient dnns. *arXiv preprint arXiv:1608.04493*.

Song Han, Jeff Pool, John Tran, and William J Dally. 2015. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*.

Wei Han, Zhengdong Zhang, Yu Zhang, Jiahui Yu, Chung-Cheng Chiu, James Qin, Anmol Gulati, Ruoming Pang, and Yonghui Wu. 2020. Contextnet: Improving convolutional neural networks for automatic speech recognition with global context. *arXiv preprint arXiv:2005.03191*.

Babak Hassibi and David G Stork. 1993. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann.

Ben Hayes, Charalampos Saitis, and György Fazekas. 2021. Neural waveshaping synthesis. *ISMIR*.

Tianxing He, Yuchen Fan, Yanmin Qian, Tian Tan, and Kai Yu. 2014. Reshaping deep neural network for fast decoding by node-pruning. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 245–249. IEEE.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021a. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *arXiv preprint arXiv:2106.07447*.

Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, et al. 2021b. Robust wav2vec 2.0: Analyzing domain shift in self-supervised pre-training. *arXiv preprint arXiv:2104.01027*.

Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021c. Hubert: How much can a bad teacher benefit asr pre-training? In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6533–6537. IEEE.

Wen-Chin Huang, Shu-Wen Yang, Tomoki Hayashi, Hung-Yi Lee, Shinji Watanabe, and Tomoki Toda. 2021. S3prl-vc: Open-source voice conversion framework with self-supervised speech representations. *arXiv preprint arXiv:2110.06280*.

Min-Jae Hwang, Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2021. Tts-by-tts: Tts-driven data augmentation for fast and high-quality speech synthesis. In *ICASSP*.

Keith Ito and Linda Johnson. 2017. The lj speech dataset. `https://keithito.com/LJ-Speech-Dataset/`.

Dongwei Jiang, Wubo Li, Miao Cao, Ruixiong Zhang, Wei Zou, Kun Han, and Xiangang Li. 2020. Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning. *arXiv preprint arXiv:2010.13991*.

Nal Kalchbrenner, Erich Elsen, Karen Simonyan, Seb Noury, Norman Casagrande, Edward Lockhart, Florian Stimberg, Aaron Oord, Sander Dieleman, and Koray Kavukcuoglu. 2018. Efficient neural audio synthesis. In *ICML*.

Naoyuki Kanda, Guoli Ye, Yu Wu, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, and Takuya Yoshioka. 2021. Large-scale pre-training of end-to-end multi-talker asr for meeting transcription with single distant microphone. *arXiv preprint arXiv:2103.16776*.

Eugene Kharitonov, Ann Lee, Adam Polyak, Yossi Adi, Jade Copet, Kushal Lakhotia, Tu-Anh Nguyen, Morgane Rivière, Abdelrahman Mohamed, Emmanuel Dupoux, et al. 2021. Text-free prosody-aware generative spoken language modeling. *arXiv preprint arXiv:2109.03264*.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. 2020a. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *NeurIPS*.

Qiuqiang Kong, Yin Cao, Turab Iqbal, Yuxuan Wang, Wenwu Wang, and Mark D Plumbley. 2020b. Panns: Large-scale pretrained audio neural networks for audio pattern recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2880–2894.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. 2021. Diffwave: A versatile diffusion model for audio synthesis. *ICLR*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of bert. *arXiv preprint arXiv:1908.08593*.

Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. 2019. Melgan: Generative adversarial networks for conditional waveform synthesis. *NeurIPS*.

Cheng-I Lai. 2019. Contrastive predictive coding based feature for automatic speaker verification. *arXiv preprint arXiv:1904.01575*.

Cheng-I Lai, Yung-Sung Chuang, Hung-Yi Lee, Shang-Wen Li, and James Glass. 2021a. Semi-supervised spoken language understanding via self-supervised speech and language model pretraining. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7468–7472. IEEE.

Cheng-I Jeff Lai, Yang Zhang, Alexander H Liu, Shiyu Chang, Yi-Lun Liao, Yung-Sung Chuang, Kaizhi Qian, Sameer Khurana, David Cox, and James Glass. 2021b. Parp: Prune, adjust and re-prune for self-supervised speech recognition. *NeurIPS*.

Kushal Lakhotia, Evgeny Kharitonov, Wei-Ning Hsu, Yossi Adi, Adam Polyak, Benjamin Bolte, Tu-Anh Nguyen, Jade Copet, Alexei Baevski, Adelrahman Mohamed, et al. 2021. Generative spoken language modeling from raw audio. *arXiv preprint arXiv:2102.01192*.

Yann LeCun, John S Denker, and Sara A Solla. 1990. Optimal brain damage. In *Advances in neural information processing systems*, pages 598–605.

Ann Lee, Peng-Jen Chen, Changhan Wang, Jiatao Gu, Xutai Ma, Adam Polyak, Yossi Adi, Qing He, Yun Tang, Juan Pino, et al. 2021. Direct speech-to-speech translation with discrete units. *arXiv preprint arXiv:2107.05604*.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*.

Bo Li, Ruoming Pang, Tara N Sainath, Anmol Gulati, Yu Zhang, James Qin, Parisa Haghani, W Ronny Huang, and Min Ma. 2021a. Scaling end-to-end models for large-scale multilingual asr. *arXiv preprint arXiv:2104.14830*.

Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. 2016. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*.

Naihan Li, Shujie Liu, Yanqing Liu, Sheng Zhao, and Ming Liu. 2019. Neural speech synthesis with transformer network. In *AAAI*.

Shengqiang Li, Menglong Xu, and Xiao-Lei Zhang. 2021b. Efficient conformer-based speech recognition with linear attention. *arXiv preprint arXiv:2104.06865*.

Shaoshi Ling and Yuzong Liu. 2020. Decoar 2.0: Deep contextualized acoustic representations with vector quantization. *arXiv preprint arXiv:2012.06659*.

Alexander H Liu, Yu-An Chung, and James Glass. 2020a. Non-autoregressive predictive coding for learning speech representations from local dependencies. *arXiv preprint arXiv:2011.00406*.

Andy T Liu, Shang-Wen Li, and Hung-yi Lee. 2021. Tera: Self-supervised learning of transformer encoder representation for speech. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2351–2366.

Andy T Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020b. Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6419–6423. IEEE.

Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. 2018. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*.

Renqian Luo, Xu Tan, Rui Wang, Tao Qin, Jinzhu Li, Sheng Zhao, Enhong Chen, and Tie-Yan Liu. 2021. Lightspeech: Lightweight and fast text to speech with neural architecture search. In *ICASSP*.

Takashi Maekaku, Xuankai Chang, Yuya Fujita, Li-Wei Chen, Shinji Watanabe, and Alexander Rudnicky. 2021. Speech representation learning combining conformer cpc with deep cluster for the zerospeech challenge 2021. *arXiv preprint arXiv:2107.05899*.

Eran Malach, Gilad Yehudai, Shai Shalev-Schwartz, and Ohad Shamir. 2020. Proving the lottery ticket hypothesis: Pruning is all you need. In *International Conference on Machine Learning*, pages 6682–6691. PMLR.

Yen Meng, Yi-Hui Chou, Andy T Liu, and Hung-yi Lee. 2021. Don't speak too fast: The impact of data bias on self-supervised speech models. *arXiv preprint arXiv:2110.07957*.

Chenfeng Miao, Shuang Liang, Minchuan Chen, Jun Ma, Shaojun Wang, and Jing Xiao. 2020. Flow-tts: A non-autoregressive network for text to speech based on flow. In *ICASSP*.

Chenfeng Miao, Liang Shuang, Zhengchen Liu, Chen Minchuan, Jun Ma, Shaojun Wang, and Jing Xiao. 2021. Efficienttts: An efficient and high-quality text-to-speech architecture. In *ICML*.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272.

Rajiv Movva and Jason Y Zhao. 2020. Dissecting lottery ticket transformers: Structural and behavioral study of sparse neural machine translation. *arXiv preprint arXiv:2009.13270*.

Jesse Mu and Jacob Andreas. 2020. Compositional explanations of neurons. *arXiv preprint arXiv:2006.14032*.

Sharan Narang, Erich Elsen, Gregory Diamos, and Shubho Sengupta. 2017. Exploring sparsity in recurrent neural networks. *arXiv preprint arXiv:1704.05119*.

Edwin G Ng, Chung-Cheng Chiu, Yu Zhang, and William Chan. 2021. Pushing the limits of non-autoregressive speech recognition. *arXiv preprint arXiv:2104.03416*.

Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. 2018a. Parallel wavenet: Fast high-fidelity speech synthesis. In *ICML*.

Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. 2016. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018b. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *ICASSP*.

Sankaran Panchapagesan, Daniel S Park, Chung-Cheng Chiu, Yuan Shangguan, Qiao Liang, and Alexander Gruenstein. 2021. Efficient knowledge distillation for rnn-transducer models. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5639–5643. IEEE.

Kainan Peng, Wei Ping, Zhao Song, and Kexin Zhao. 2020. Non-autoregressive neural text-to-speech. In *ICML*.

Wei Ping, Kainan Peng, and Jitong Chen. 2019. Clarinet: Parallel wave generation in end-to-end text-to-speech. *ICLR*.

Wei Ping, Kainan Peng, Kexin Zhao, and Zhao Song. 2020. Waveflow: A compact flow-based model for raw audio. In *ICML*.

Adam Polyak, Yossi Adi, Jade Copet, Eugene Kharitonov, Kushal Lakhotia, Wei-Ning Hsu, Abdelrahman Mohamed, and Emmanuel Dupoux. 2021. Speech resynthesis from discrete disentangled self-supervised representations. *arXiv preprint arXiv:2104.00355*.

Daniel Povey, Gaofeng Cheng, Yiming Wang, Ke Li, Hainan Xu, Mahsa Yarmohammadi, and Sanjeev Khudanpur. 2018. Semi-orthogonal low-rank matrix factorization for deep neural networks. In *Interspeech*, pages 3743–3747.

Sai Prasanna, Anna Rogers, and Anna Rumshisky. 2020. When bert plays the lottery, all tickets are winning. *arXiv preprint arXiv:2005.00561*.

Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020a. Massively multilingual asr: 50 languages, 1 model, 1 billion parameters. *arXiv preprint arXiv:2007.03001*.

Vineel Pratap, Qiantong Xu, Jacob Kahn, Gilad Avidov, Tatiana Likhomanenko, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. 2020b. Scaling up online speech recognition using convnets. *arXiv preprint arXiv:2001.09727*.

Ryan Prenger, Rafael Valle, and Bryan Catanzaro. 2019. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP*.

Janne Pylkkönen. 2005. New pruning criteria for efficient decoding. In *Ninth European Conference on Speech Communication and Technology*.

Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. 2020. What's hidden in a randomly weighted neural network? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11893–11902.

Yi Ren, Chenxu Hu, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2021. Fastspeech 2: Fast and high-quality end-to-end text to speech. *ICLR*.

Yi Ren, Yangjun Ruan, Xu Tan, Tao Qin, Sheng Zhao, Zhou Zhao, and Tie-Yan Liu. 2019. Fastspeech: Fast, robust and controllable text to speech. *NeurIPS*.

Alex Renda, Jonathan Frankle, and Michael Carbin. 2020. Comparing rewinding and fine-tuning in neural network pruning. *arXiv preprint arXiv:2003.02389*.

Morgane Rivière, Armand Joulin, Pierre-Emmanuel Mazaré, and Emmanuel Dupoux. 2020. Unsupervised pretraining transfers well across languages. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7414–7418. IEEE.

Aaqib Saeed, David Grangier, and Neil Zeghidour. 2021. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3875–3879. IEEE.

Ramon Sanabria, Austin Waters, and Jason Baldridge. 2021. Talk, don't write: A study of direct speech-based image retrieval. *arXiv preprint arXiv:2104.01894*.

Victor Sanh, Thomas Wolf, and Alexander M Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. *arXiv preprint arXiv:2005.07683*.

Yuan Shangguan, Jian Li, Qiao Liang, Raziel Alvarez, and Ian McGraw. 2019. Optimizing speech recognition for the edge. *arXiv preprint arXiv:1909.12408*.

Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. 2018. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *ICASSP*.

Yangyang Shi, Yongqiang Wang, Chunyang Wu, Ching-Feng Yeh, Julian Chan, Frank Zhang, Duc Le, and Mike Seltzer. 2021. Emformer: Efficient memory transformer based acoustic model for low latency streaming speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6783–6787. IEEE.

Vesa Siivola, Teemu Hirsimaki, and Sami Virpioja. 2007. On growing and pruning kneser–ney smoothed $n$-gram models. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(5):1617–1624.

Gabriel Synnaeve, Qiantong Xu, Jacob Kahn, Tatiana Likhomanenko, Edouard Grave, Vineel Pratap, Anuroop Sriram, Vitaliy Liptchinsky, and Ronan Collobert. 2019. End-to-end asr: from supervised to semi-supervised learning with modern architectures. *arXiv preprint arXiv:1911.08460*.

Xu Tan, Tao Qin, Frank Soong, and Tie-Yan Liu. 2021. A survey on neural speech synthesis. *arXiv preprint arXiv:2106.15561*.

Liang-Hsuan Tseng, Yu-Kuan Fu, Heng-Jui Chang, and Hung-yi Lee. 2021. Mandarin-english code-switching speech recognition with self-supervised speech representation models. *arXiv preprint arXiv:2110.03504*.

Hugo Van Hamme and Filip Van Aelten. 1996. An adaptive-beam pruning technique for continuous speech recognition. In *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP'96*, volume 4, pages 2083–2086. IEEE.

Ganesh Venkatesh, Alagappan Valliappan, Jay Mahadeokar, Yuan Shangguan, Christian Fuegen, Michael L Seltzer, and Vikas Chandra. 2021. Memory-efficient speech recognition on smart devices. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8368–8372. IEEE.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned. *arXiv preprint arXiv:1905.09418*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.

Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Yao Qian, Kenichi Kumatani, and Furu Wei. 2021a. Unispeech at scale: An empirical study of pre-training method on large-scale speech recognition dataset. *arXiv preprint arXiv:2107.05233*.

Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang. 2021b. Unispeech: Unified speech representation learning with labeled and unlabeled data. *arXiv preprint arXiv:2101.07597*.

Jun Wang, Max W Y Lam, Dan Su, and Dong Yu. 2021c. Contrastive separative coding for self-supervised representation learning. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3865–3869. IEEE.

Weiran Wang, Qingming Tang, and Karen Livescu. 2020. Unsupervised pre-training of bidirectional speech encoders via masked reconstruction. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6889–6893. IEEE.

Yiming Wang, Jinyu Li, Heming Wang, Yao Qian, Chengyi Wang, and Yu Wu. 2021d. Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition. *arXiv preprint arXiv:2110.04934*.

Matthew Wiesner, Desh Raj, and Sanjeev Khudanpur. 2021. Injecting text and cross-lingual supervision in few-shot learning from self-supervised models. *arXiv preprint arXiv:2110.04863*.

Zhaofeng Wu, Ding Zhao, Qiao Liang, Jiahui Yu, Anmol Gulati, and Ruoming Pang. 2021. Dynamic sparsity neural networks for automatic speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6014–6018. IEEE.

Ji Xin, Jimmy Lin, and Yaoliang Yu. 2019. What part of the neural network does this? understanding lstms by measuring and dissecting neurons. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5827–5834.

Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. 2018. A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5929–5933. IEEE.

Qiantong Xu, Alexei Baevski, and Michael Auli. 2021a. Simple and effective zero-shot cross-lingual phoneme recognition. *arXiv preprint arXiv:2109.11680*.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021b. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034. IEEE.

Jian Xue, Jinyu Li, and Yifan Gong. 2013. Restructuring of deep neural network acoustic models with singular value decomposition. In *Interspeech*, pages 2365–2369.

Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. 2020. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. 2021. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*.

Zhao You, Shulin Feng, Dan Su, and Dong Yu. 2021. Speechmoe: Scaling to large acoustic models with dynamic routing mixture of experts. *arXiv preprint arXiv:2105.03036*.

Dong Yu, Frank Seide, Gang Li, and Li Deng. 2012. Exploiting sparseness in deep neural networks for large vocabulary speech recognition. In *2012 IEEE International conference on acoustics, speech and signal processing (ICASSP)*, pages 4409–4412. IEEE.

Haonan Yu, Sergey Edunov, Yuandong Tian, and Ari S Morcos. 2019. Playing the lottery with rewards and multiple languages: lottery tickets in rl and nlp. *arXiv preprint arXiv:1906.02768*.

Jiahui Yu, Wei Han, Anmol Gulati, Chung-Cheng Chiu, Bo Li, Tara N Sainath, Yonghui Wu, and Ruoming Pang. 2020. Universal asr: Unify and improve streaming asr with full-context modeling. *arXiv preprint arXiv:2010.06030*.

Jiahui Yu, Wei Han, Anmol Gulati, Chung-Cheng Chiu, Bo Li, Tara N Sainath, Yonghui Wu, and Ruoming Pang. 2021. Dual-mode asr: Unify and improve streaming asr with full-context modeling. *Proceedings of ICLR*.

Heiga Zen, Keiichi Tokuda, and Alan W Black. 2009. Statistical parametric speech synthesis. *speech communication*.

Bohan Zhai, Tianren Gao, Flora Xue, Daniel Rothchild, Bichen Wu, Joseph E Gonzalez, and Kurt Keutzer. 2020. Squeezewave: Extremely lightweight vocoders for on-device speech synthesis. *arXiv preprint arXiv:2001.05685*.

Yu Zhang, Daniel S Park, Wei Han, James Qin, Anmol Gulati, Joel Shor, Aren Jansen, Yuanzhong Xu, Yanping Huang, Shibo Wang, et al. 2021a. Bigssl: Exploring the frontier of large-scale semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2109.13226*.

Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu. 2020. Pushing the limits of semi-supervised learning for automatic speech recognition. *arXiv preprint arXiv:2010.10504*.

Yuekai Zhang, Sining Sun, and Long Ma. 2021b. Tiny transducer: A highly-efficient speech recognition model on edge devices. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6024–6028. IEEE.

Han Zhu, Li Wang, Ying Hou, Jindong Wang, Gaofeng Cheng, Pengyuan Zhang, and Yonghong Yan. 2021. Wav2vec-s: Semi-supervised pre-training for speech recognition. *arXiv preprint arXiv:2110.04484*.