PARP: Prune, Adjust and Re-Prune for Self-Supervised Speech Recognition

Cheng-I Jeff Lai¹, Yang Zhang², Alexander H. Liu¹, Shiyu Chang², Yi-Lun Liao³, Yung-Sung Chuang³, Kaizhi Qian², Sameer Khurana¹
David Cox², James Glass³

¹MIT CSAIL, ²MIT-IBM Watson AI Lab, ³National Taiwan University

Abstract

Recent work on speech self-supervised learning (speech SSL) demonstrated the benefits of scale in learning rich and transferable representations for Automatic Speech Recognition (ASR) with limited parallel data. It is then natural to investigate the existence of sparse and transferrable subnetworks in pre-trained speech SSL models that can achieve even better low-resource ASR performance. However, directly applying widely adopted pruning methods such as the Lottery Ticket Hypothesis (LTH) is suboptimal in the computational cost needed. Moreover, contrary to what LTH predicts, the discovered subnetworks yield minimal performance gain compared to the original dense network. In this work, we propose Prune-Adjust-Re-Prune (PARP), which discovers and finetunes subnetworks for much better ASR performance, while only requiring a single downstream finetuning run. PARP is inspired by our surprising observation that subnetworks pruned for pre-training tasks only needed to be slightly adjusted to achieve a sizeable performance boost in downstream ASR tasks. Extensive experiments on low-resource English and multi-lingual ASR show (1) sparse subnetworks exist in pre-trained speech SSL, and (2) the computational advantage and performance gain of PARP over baseline pruning methods. On the 10min Librispeech split without LM decoding, PARP discovers subnetworks from wav2vec 2.0 with an absolute 10.9%/12.6% WER decrease compared to the full model. We demonstrate PARP mitigates performance degradation in cross-lingual mask transfer, and investigate the possibility of discovering a single subnetwork for 10 spoken languages in one run.

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 [20, 22, 31, 21, 16]. Recently, Self-Supervised Speech Representation Learning (speech SSL) has emerged as a promising pathway toward solving low-resource ASR [4, 19, 93, 42]. 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 [4, 19, 93]. Prior research attempts, however, focus on pre-training objectives [60, 18, 82, 42], scaling up speech representation modules [3, 4], or pre-training data selections [41]. 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.

*Equal contribution.
We conduct extensive experiments with mono-lingual (pre-trained wav2vec 2.0 [4]) and cross-lingual (pre-trained XLSR-53 [19]) SSL with improved performance on low-resource ASR? How can we find sparse subnetworks within pre-trained speech SSL that can achieve superior performance to the full pre-trained model on downstream ASR tasks.  

However, directly applying widely-adopted pruning methods, such as One-Shot Magnitude Pruning (OMP) and Iterative Magnitude Pruning (IMP) [37,28], to pre-trained speech SSL suffers from two challenges. First, adopting these methods in the conventional pruning framework is extremely time-consuming for SOTA speech SSL models. OMP and IMP involve more than one round of finetuning on downstream tasks (c.f. Figure 1), and finetuning for ASR is time-consuming and computationally demanding [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 5 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 pre-trained speech SSL with improved performance on low-resource ASR? How can we discover them efficiently in a single downstream finetuning run? 

In this paper, we propose a magnitude-based unstructured pruning method [30,9], 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. Different from pruning methods in [37,28], PARP allows pruned-out weights to be revived during finetuning. Although such a high-level idea was introduced in [56], 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 in Section 3 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 mask transfer experiments, where the initial subnetwork is obtained via a different language other than the target language.

We conduct extensive PARP and baseline (OMP and IMP) pruning experiments on low-resource ASR with mono-lingual (pre-trained wav2vec 2.0 [4]) and cross-lingual (pre-trained XLSR-53 [19]) transfer. PARP finds significantly superior speech SSL subnetworks for low-resource ASR, while only requiring a single pass of downstream finetuning. Due to its simplicity, PARP adds minimal computation overhead to existing SSL downstream finetuning. Our contributions are:  

- We show that sparse subnetworks exist in pre-trained speech SSL. 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. On Librispeech 10min without

---

2Standard wav2vec 2.0 finetuning setup [4] 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 BERT on GLUE [60].
We adopted pre-trained speech SSL 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 4.1).

- Ablation studies on demonstrating the importance of PARP’s initial subnetwork (Section 4.2).
- PARP achieves minimal performance drop in cross-lingual mask transfer, where a subnetwork pruned for ASR in one spoken language is adapted to another language (Section 4.3). PARP can also discover a single subnetwork for 10 spoken languages in a single pass (Section 4.4).
- Last but not least, we demonstrate PARP’s effectiveness on pre-trained BERT, mitigating the cross-task performance degradation reported in BERT-Ticket [15] (Section 4.5).

The findings of this work not only complement and advance current and future speech SSL for low-resource ASR, but also provide new insights for the rich body of pruning work.

2 Preliminaries

2.1 Problem Formulation

Consider the low-resource ASR problem, where there is only a small transcribed training set \((x, y) \in D_l\). 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 \(x \in D_0\). SSL pre-trains a neural network \(f(x; \theta)\), where \(\theta \in \mathbb{R}^d\) 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 \(L_l(\theta)\), such as CTC, and evaluated on target dataset \(D_l\).

Our goal is to discover a subnetwork that minimizes downstream ASR WER on \(D_l\). 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 \(D_l\). 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 \(L_l(\theta)\) loss on \(D_l\).

2.2 Pruning Targets and Settings

We adopted pre-trained speech SSL wav2vec2 and x1sr 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 [4]. During finetuning, a task specific linear layer is added on top of wav2vec2 and jointly finetuned with CTC loss. More details can be found in Appendix A.

**XLSR-53 (x1sr)** shares the same architecture, pre-training and finetuning objectives as wav2vec2-large. x1sr 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 x1sr 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 x1sr is finetuned for low-resource multi-lingual ASR.

2.3 Subnetwork Discovery in Pre-trained SSL

One obvious solution to the aforementioned problem in Section 2.1 is to directly apply pruning with rewinding to \(\theta_0\), which has been successfully applied to pre-trained BERT [15] and SimCLR [14]. All pruning methods, including our proposed PARP, are based on Unstructured Magnitude Pruning (UMP) [28, 30], 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) [28, 30, 15], 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 $D_l$ 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 [15], 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 Initializations (MPI). Another case is pruning weights finetuned for a different language $l$, i.e. applying UMP on $\theta^l$ 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, i.e. finetune $f(x; m \odot \theta_0)$ on $D_l$.

### 3 Method

In this section, we highlight our proposed pruning method, PARP (Section 3.1), its underlying intuition (Section 3.2), and an extension termed PARP-P (Section 3.3).

#### 3.1 Algorithm

We formally describe PARP with the notations from Section 2. A visual overview of PARP is Figure 2.

**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 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 $\theta_{a1}$ given by $m$. Lift up $m$ such that whole $\theta_{a1}$ is updatable.
   5. Train $f(x; \theta_{a1})$ for $n$ model updates and obtain $f(x; \theta_{a2})$.
   6. Apply UMP on $f(x; \theta_{a2})$ and adjust $m$ accordingly. The adjusted subnetwork is $f(x; m \odot \theta_{a2})$. 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.

#### 3.2 Obtaining and Adjusting the Initial Subnetwork

PARP achieves superior or comparable pruning results as task-aware subnetwork discovery, 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:

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

Take H2L and CSR for instance, Figure 2 visualizes language pairs’ OMP pruning mask IOUs on wav2vec2 and x1sr. Observe the high overlaps across all pairs, but also the high IOUs 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 IOUs with those obtained from task-aware subnetwork discovery.

---

3This step is referred to as subnetwork finetuning in the pruning literature [53, 69, 9].
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 [36], important weights pruning [55], and deep rewiring [8], 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.

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. 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.

Figure 2: IOUs over all spoken language pairs’ OMP pruning masks on finetuned wav2vec2 and xlsr. Second to last row is the IOUs between OMP masks and the MPI masks from pre-trained wav2vec2 and xlsr. Here, we show the IOUs at 50% sparsity, and the rest can be found in Appendix. 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. Language IDs are in Appendix B.

4 Experiments and Analysis

4.1 Comparing PARP, OMP, and IMP on LSR, H2L, and CSR

Our experimental setup can be found in Appendix B. We first investigate the existence of sparse subnetworks in speech SSL. Figure 3 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 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. The same effect is not seen with OMP, IMP, nor MPI. Table 4 compares the subnetworks discovered by PARP with the full wav2vec2 and prior work on LSR under the same setting [38]. Surprisingly, the discovered subnetwork attains an absolute 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. We also examined the effectiveness of IMP with different rewinding starting points as studied in [29], and found rewinding initializations bear minimal effect on downstream ASR. Full rewinding details are in Appendix C.

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

Table 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       |
| PARP-P 60% → 80% → 90%  | 2                 | 33.6       | 53.3       |

Table 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 + LM        | 49.5       | 66.3       |
| Discrete BERT + LM          | 16.3       | 25.2       |
| wav2vec2-base reported [4]  | 46.9       | 50.9       |
| wav2vec2-large reported [4] | 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       |

Next, we examine if the pruning results of LSR transfers to H2L and CSR. Figure 4 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 and 4, 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 4 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 compares the subnetworks from H2L and CSR with prior work. Even with as high as 90% sparsities in either settings, subnetworks from PARP and OMP out-performs prior art.

Figure 4: 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.
Table 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 [27]       | Babel-1070h 0% 44.9            |
| CPC [60]              | LS-100h 0% 50.9                |
| Modified CPC [70]     | 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 [19]    | 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               |

Figure 5: PARP’s final subnetwork and its initial MPI subnetwork exceeds 99.99% IOU after 20% sparsity (black line).

Figure 6: PARP with random (red line) v.s. with MPI (black line) initial subnetworks in LSR.

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 6 shows PARP 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, 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 5 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.

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 transferrability 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 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 7 shows the transferrability 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 induce a much higher PER compare to that from sv-SE, tr, tt, zh-TW. On the right of Figure 7, we show that there is no cross-lingual PER degradation with PARP, supporting our claim above.

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...
Figure 7: (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 better). Both are on H2L with pretrained wav2vec2. The same observation is observed on CSR with pretrained x1sr in Appendix E.

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 language, we construct a joint phoneme dictionary and finetune wav2vec2 and x1sr 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 4 illustrates the results.

Comparing joint-finetuning and individual-finetuning, in H2L, we found that the shared subnetwork obtained via OMP has lower PERs between 60~80% 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 4), its shared subnetworks are only slightly better than OMP in both H2L and CSR (right of Figure 4). 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 8 for better illustration).

4.5 Does PARP work on Pre-trained BERT?

We also analyzed whether Observation 1 holds for pre-trained BERT 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 [15] with PARP. Table 4 compares PARP (averaged for each target task) to those reported in BERT-Ticket, hinting the applicability of PARP to other domains and pre-trained models. Detailed scores and figures are in Appendix F.

Table 4: 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          | CoLA | MRPC | QNLI | QQP | RTE | SST-2 | STS-B | WNLI | MNLI |
|-----------------|------|------|------|-----|-----|-------|-------|------|------|
| Same-task Transfer (top line) | 40.65 | 76.23 | 89.36 | 91.07 | 59.93 | 90.60 | 86.85 | 56.34 | 82.87 |
| PARP            | 28.48 | 75.98 | 87.12 | 90.40 | 59.69 | 89.59 | 86.25 | 54.62 | 81.61 |
| BERT-Ticket [15]| 10.12 | 71.94 | 86.54 | 88.50 | 57.59 | 88.80 | 80.27 | 54.03 | 80.48 |

4.6 Implications

Observation 1 is consistent with the findings of probing large pre-trained NLP models such as BERT, that pre-trained SSL models are over-parametrized and there exist task-oriented weights/neurons. Figure 2 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, [26, 24, 5, 84] measured, [5, 23, 46] leveraged, [57, 34] visualized, and [79, 25, 11] 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 [7, 6, 17]. 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, see Figure 8. 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\(^6\). Finally, we make connections to prior work that showed RP prevail \([9,15,53,54,68]\) – under a certain threshold and setting, task-specific neurons are less likely to get “accidentally” pruned and thus accuracy is preserved even with RP.

5 Related Work

Modern ASR Paradigm and ASR Pruning. As model scale \([76,4,38,35,91,66,65,92,13,88,50]\) and model pre-training \([4,33,19,45,43,47,42,86,12,44,72,71,59,62,81]\) 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 \([1,40,85,94]\) and HMM state space \([77]\). Since the seminal work of Yu et al. \([89]\), ASR pruning has focused primarily on end-to-end network architecture: \([73,83]\) applied pruning and quantization to LSTM-based RNN-Transducers, \([61]\) applied knowledge distillation to Conformer-based RNN-Transducers, \([78,74,52]\) designed efficient architecture/mechanisms for LSTM, Transformer, Conformer-based ASR models, \([58]\) applied pruning to Deep Speech, \([10]\) introduced SNR-based probabilistic pruning on LSTM-based CTC model, \([80]\) proposed entropy-regularizer for LSTM-based ASR model, \([87,65]\) 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.

6 Conclusion and Broader Impact

We introduce PARP, a conceptually simple and intuitive pruning method for self-supervised speech recognition. We conduct extensive 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, and (4) PARP can discover a shared subnetwork for multiple spoken languages in one pass. The broader impact of this research work lies 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 SSL frameworks that reduces the computational costs required for adapting (finetuning) them to custom settings. We do not see its potential societal harm.

\(^6\)In CV, it is standard to iterate the process hundreds of times \([36]\).
References

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

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

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

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

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

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

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

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

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

[10] Stefan Braun and Shih-Chii Liu. 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, 2019.

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

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

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

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

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

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

[17] Yu-An Chung, Yonatan Belinkov, and James Glass. 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, 2021.

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

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

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

[21] Jia Cui, Brian Kingsbury, Bhuvana Ramabhadran, Abhinav Sethy, Kartik Audhkhasi, Xiaodong Cui, Ellen Kislar, Lidia Mangu, Markus Nussbaum-Thom, Michael Picheny, et al. 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, 2015.
[22] Xiaodong Cui, Brian Kingsbury, Jia Cui, Bhuvana Ramabhadran, Andrew Rosenberg, Mohammad Sadegh Rasooli, Owen Rambow, Nizar Habash, and Vaibhava Goel. 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, 2014.

[23] Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, and Furu Wei. Knowledge neurons in pretrained transformers. arXiv preprint arXiv:2010.02695, 2020.

[24] Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 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, 2019.

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

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

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

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

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

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

[31] Mark JF Gales, Kate M Knill, Anton Ragni, and Shakti P Rath. 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), 2014.

[32] Dawei Gao, Xiaoxi He, Zimu Zhou, Yongxin Tong, Ke Xu, and Lothar Thiele. 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, 2020.

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

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

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

[36] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for fast decoding by node-pruning. In 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 245–249. IEEE, 2014.

[37] Wei-Ning Hsu, Yao-Hung Hubert Tsai, Benjamin Bolte, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 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, 2021.
[43] Dongwei Jiang, Wubo Li, Miao Cao, Ruixiong Zhang, Wei Zou, Kun Han, and Xiangang Li. Speech simclr: Combining contrastive and reconstruction objective for self-supervised speech representation learning. arXiv preprint arXiv:2010.13991, 2020.

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

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

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

[47] Cheng-I Lai, Yung-Sung Chuang, Hung-Yi Lee, Shang-Wen Li, and James Glass. 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, 2021.

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

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

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

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

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

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

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

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

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

[57] Jesse Mu and Jacob Andreas. Compositional explanations of neurons. arXiv preprint arXiv:2006.14032, 2020.

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

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

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

[61] Sankaran Panchapagesan, Daniel S Park, Chung-Cheng Chiu, Yuan Shangguan, Qiao Liang, and Alexander Gruenstein. 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, 2021.

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

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

[64] Sai Prasanna, Anna Rogers, and Anna Rumshisky. When bert plays the lottery, all tickets are winning. arXiv preprint arXiv:2005.00561, 2020.
[65] Vineel Pratap, Anuroop Sriram, Paden Tomasello, Awni Hannun, Vitaliy Liptchinsky, Gabriel Synnaeve, and Ronan Collobert. Massively multilingual asr: 50 languages, 1 model, 1 billion parameters. *arXiv preprint arXiv:2007.03001*, 2020.

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

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

[68] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. 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, 2020.

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

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

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

[72] Jun Wang, Max W Y Lam, Dan Su, and Dong Yu. Contrastive separative coding for self-supervised representation learning. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6889–6893. IEEE, 2020.

[73] Zhaofeng Wu, Ding Zhao, Qiao Liang, Jiahui Yu, Anmol Gulati, and Ruoming Pang. 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, 2021.
[85] Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. 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, 2018.

[86] Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasetto, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 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, 2021.

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

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

[89] Dong Yu, Frank Seide, Gang Li, and Li Deng. 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, 2012.

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

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

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

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

[94] Yuekai Zhang, Sining Sun, and Long Ma. 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, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper’s contributions and scope? [Yes] Through experiments results are in Section 4. For example, our claim that PARP outperforms LTH is visible in Figure 3.
   (b) Did you describe the limitations of your work? [No] Not in the main content, but we include it in the Appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [No] We mention in Section 6 on the broader impact of this research work. Since this work is on pruning existing speech SSL models for low-resource spoken languages, we do not see its potential negative societal impacts. However, we welcome reviewers and AC to raise such concerns, and we will include corresponding statements.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]

3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Due to its simplicity, PARP only adds a few lines of code to. Data and pre-trained models are all publicly available. These details are in the Appendix.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We follow [4, 19] for the fine-tuning hyper-parameters. These details are in the Appendix.
(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] Due to the computational expense and scale of our experiments, we were not able to extensively re-run. We do note that our re-created baselines match the numbers reported in prior work [4, 19].

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We briefly mention the compute needed in the footnote in Page 2, and more details are in the Appendix.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

   (a) If your work uses existing assets, did you cite the creators? [Yes] Our work (code and pre-trained models) are based on [4, 19].

   (b) Did you mention the license of the assets? [N/A]

   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]

   (d) Did you use crowdsourcing or conducted research with human subjects... [N/A]

   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] We used published datasets and, to the best of our knowledge, all of them have been reviewed carefully by the authors/community.

5. If you used crowdsourcing or conducted research with human subjects...

   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]
Appendix

A Model Details

Model configurations for wav2vec2-base, wav2vec2-large, and xlsr can be found in Section A.1. Finetuning hyper-parameters are generally the same as in [4], and we detailed them in Section A.2. PARP’s hyper-parameter is detailed in Section A.3. More details on system implementations is in Section A.4.

A.1 Model Configurations

wav2vec 2.0 consists of three modules: a 7-layer CNN feature encoder for pre-processing raw speech waveforms, a quantization layer for discretizing, and a BERT for learning contextualized representations. Given that the feature encoder is fixed and the quantization layer is discarded during finetuning, we focus on pruning the BERT module in wav2vec 2.0 and XLSR-53. We also do not prune the positional embedding layer nor the layer normalization layers within BERT. This setup is consistent with BERT-Ticket [15]. wav2vec 2.0 BASE (wav2vec2-base) is based on BERT-BASE, which has 12 transformer blocks, hidden dimension 768, 12 self-attention heads, and 95M parameters. wav2vec 2.0 LARGE (denote as wav2vec2-large) is based on BERT-LARGE, which has 24 transformer blocks, hidden dimension 768, 16 self-attention heads, and 315M parameters. XLSR-53 (denoted as xlsr) shares the same architecture as wav2vec2-large. We took wav2vec2-base and wav2vec2-large that were pre-trained on Librispeech 960h. wav2vec2-base, wav2vec2-large, and xlsr are pre-trained with the contrastive predictive coding objective.

A.2 Finetuning Hyper-Parameters

wav2vec2 is finetuned for 20k steps on the 10h split, 15k steps on the 1h split, and 12k steps on the 10min split. xlsr is finetuned for 12k steps for each spoken languages. In the default setup in [4], wav2vec2 except the final linear layer is freeze for 10k steps, however, we observe doing so on the pruned models may lead to training instability. Therefore, we do not include this trick in our fine-tuning setups. The learning rate ramps up linearly for first 10% of the steps, remains the same for 40% of the steps, and decay exponentially for 50% of the steps. The waveform encoder output is randomly masked according to [4]. For LSR, the validation set is the dev-other subset from Librispeech.

A.3 PARP Hyper-Parameters

PARP introduces an additional pruning frequency hyper-parameter, $n$ in Algorithm Table 1. As long as $n$ is a sensible small number (e.g. 5-50 out of 10k+ steps), the final pruned models should have similar performance. We heuristically set $n = 5$ for pruning XLSR on all spoken language splits; we set $n = 50$ for wav2vec2-base on 10min/1h, $n = 5$ for wav2vec2-base on 10h, $n = 5$ for wav2vec2-large on 10min, $n = 2$ for wav2vec2-large on 1h, and $n = 1$ for wav2vec2-large on 10h.

A.4 Implementation

All experiments are based on the Fairseq repository [7] and Wav2letter++ decoding [8]. We took publicly available pre-trained wav2vec2-base, wav2vec2-large, and xlsr [9]. The pruning code is based on PyTorch’s pruning module [10]. For each experiment, we fine-tune the model on either 2 or 4 GPUs in parallel, and unlike the standard wav2vec 2.0 fine-tuning setup, we do not include a LM for validation during fine-tuning. Given that not all of our GPUs support FP16, our fine-tuning setup is on FP32. For fair comparison, we imposed a reasonable computational budget for all pruning methods used in this study[11].

[7] https://github.com/pytorch/fairseq
[8] https://github.com/flashlight/wav2letter
[9] Pre-trained models available at https://github.com/pytorch/fairseq/blob/master/examples/wav2vec/README.md
[10] https://pytorch.org/tutorials/intermediate/pruning_tutorial.html
[11] Each finetuning run is capped at a total of 100 V100 hours. For example, OMP requires 2 finetunings, so we will run it for at most a total of 50 hours on across 4 V100s.
### B Experimental Setup for LSR, H2L, and CSR

For LSR, we finetune pre-trained `wav2vec2-base` and `wav2vec2-large` on the 10h/1h/10min splits from Librispeech and Libri-light, as this is the *de facto* setup for studying speech representation learning [4]. For H2L, we replicate the setting described in [70, 19], where pre-trained `wav2vec2-base` is finetuned on 10 spoken languages (1 hour each) from CommonVoice: Spanish (es), French (fr), Italian (it), Kyrgyz (ky), Dutch (nl), Russian (ru), Swedish (sv-SE), Turkish (tr), Tatar (tt), and Mandarin (zh-TW). For CSR, we replicate the setting in [19], where pre-trained `xlsr` is finetuned on the same 10 languages as in H2L. Studying LSR can inform us the effect of amount of finetuning supervision (10min∼10h) and pre-trained model scales (`base` v.s. `large`) on pruning; on the other hand, comparing CSR and H2L could yield insights on the effect of mono-lingual versus cross-lingual pre-training on pruning.

**Evaluation Criteria.** Word Error Rate (WER) is reported for LSR; Phone Error Rate (PER) is reported for H2L and CSR\(^{12}\). Earlier work on pruning sequence to sequence tasks, such as ASR [10] or Machine Translation [90, 30], showed that pruned models do not match or outperform the full model, albeit with “minimal degradation”. Moreover, to isolate the effects of different pruning methods, we do not include any external LM nor any means of self-training [86] during training or decoding. To provide an unbiased grounding and accurate reflection of the pruned models, we thus report relative gains of our proposed method over OMP/IMP/MPI, in addition to their raw WER/PERs.

### C How important is the IMP rewinding starting point?

We also examined the effectiveness of IMP rewinding [29, 69] for pruning speech SSL, where instead of re-starting each IMP pruning iteration all the way back from pre-trained SSL initializations, the iteration starts at some points during the downstream ASR finetuning. For example, in figure 9, IMP with 10% rewinding (dark red line) means that each pruning iteration starts at 10% into the ASR downstream finetuning; We find that rewinding has minimal effect for pruning speech SSL, which aligns with the results in NLP [15]. Curiously, we observe the effect diminishes when the pre-training model size is scaled up from `base` to `large`.

![Figure 9: IMP on `wav2vec2-base` and `wav2vec2-large` with different rewinding starting point within the downstream ASR finetuning. Its effect diminishes when pruning `wav2vec2-large`.

---

\(^{12}\)WER/PER (lower the better) is standard criteria for ASR. This is opposite to previous work on pruning CV or NLP models, where accuracy or BLEU scores (higher the better) was reported.
In this section, we provide the rest of Figure 2 at other sparsities, to support Observation 1. In addition to IOU, we also provide the overlap percentage between masks. We divide this section into OMP masks overlap over spoken language pairs on finetuned wav2vec2-base in H2L (Section D.1) and overlaps on finetuned xlsr in CSR (Section D.2).

### D.1 OMP Masks Overlap in H2L

![Figure 10: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 10% sparsity.](image1)

![Figure 11: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 20% sparsity.](image2)

---

13 For readability, we re-state Observation here: 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 IOUs with those obtained from task-aware subnetwork discovery.

14 Instead of taking the Union in the denominator as in IOU, simply take the full number of parameters.
Figure 12: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 30% sparsity.

Figure 13: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 40% sparsity.

Figure 14: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 50% sparsity.
Figure 15: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 60% sparsity.

Figure 16: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 70% sparsity.

Figure 17: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 80% sparsity.
Figure 18: OMP pruning masks IOUs and overlap percentages on finetuned wav2vec2 at 90% sparsity.

D.2 OMP Masks Overlap in CSR

Figure 19: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 10% sparsity.

Figure 20: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 20% sparsity.
Figure 21: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 30% sparsity.

Figure 22: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 40% sparsity.

Figure 23: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 50% sparsity.
Figure 24: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 60% sparsity.

Figure 25: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 70% sparsity.

Figure 26: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 80% sparsity.
Figure 27: OMP pruning masks IOUs and overlap percentages on finetuned xlsr at 90% sparsity.

**E**  **xlsr Cross-Lingual Mask Transfer**

Figure 28: Cross-lingual mask transfer for pretrained xlsr. Cross-lingual mask transfer with PARP has minimal PER degradation (darker the better).
Details of Task Transfer Results on Pre-trained BERT

For all the GLUE tasks, PARP can achieve better results compared to BERT-Ticket [15]. For the tasks with poor transferability in BERT-Ticket [15], like CoLA and STS-B, PARP can still achieve good transfer scores.

Figure 29: Results for subnetwork transfer experiment (take subnetwork found by IMP at task A and finetune it for task B). Left: the transfer results in BERT-Ticket [15]. Right: transfer with PARP finetuning instead. Each row is a source task A, and each column is a target task B. All numbers are subtracted by the scores of same-task transfer (task A = task B, and the darker the better).

Figure 30: IOU of GLUE tasks' IMP pruning masks on finetuned BERT at 70% sparsity. Notice the high overlap rates, which aligns with Observation 1.
G Full H2L and CSR Pruning Results

Figure 31: Comparison of pruning techniques on H2L & CSR with 1h of Spanish (es) ASR finetuning. (Left) Pruning H2L (wav2vec2-base + es). (Center) Pruning CSR (xlsr + es). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on es.

Figure 32: Comparison of pruning techniques on H2L & CSR with 1h of French (fr) ASR finetuning. (Left) Pruning H2L (wav2vec2-base + fr). (Center) Pruning CSR (xlsr + fr). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on fr.

Figure 33: Comparison of pruning techniques on H2L & CSR with 1h of Italian (it) ASR finetuning. (Left) Pruning H2L (wav2vec2-base + it). (Center) Pruning CSR (xlsr + it). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on it.
Figure 34: Comparison of pruning techniques on H2L & CSR with 1h of Kyrgyz (ky) ASR finetuning. 
(Left) Pruning H2L (wav2vec2-base + ky). (Center) Pruning CSR (xlsr + ky). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on ky.

Figure 35: 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.

Figure 36: Comparison of pruning techniques on H2L & CSR with 1h of Russian (ru) ASR finetuning. 
(Left) Pruning H2L (wav2vec2-base + ru). (Center) Pruning CSR (xlsr + ru). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on ru.
Figure 37: Comparison of pruning techniques on H2L & CSR with 1h of Swedish (sv-SE) ASR finetuning. (Left) Pruning H2L (wav2vec2-base + sv-SE). (Center) Pruning CSR (xlsr + sv-SE). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on sv-SE.

Figure 38: Comparison of pruning techniques on H2L & CSR with 1h of Turkish (tr) ASR finetuning. (Left) Pruning H2L (wav2vec2-base + tr). (Center) Pruning CSR (xlsr + tr). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on tr.

Figure 39: Comparison of pruning techniques on H2L & CSR with 1h of Tatar (tt) ASR finetuning. (Left) Pruning H2L (wav2vec2-base + tt). (Center) Pruning CSR (xlsr + tt). (Right) Pruning jointly-finetuned wav2vec2-base and xlsr on tt.
**H Limitations**

Below, we list several limitations of PARP and the experimental designs presented in this paper:

1. Experiments are on contrastive pre-trained models only. It is unclear whether the results would generalize to pre-trained models with other objectives, such as mask prediction (HuBERT) or autoregressive prediction (APC), etc.

2. Although standard, our experiments are on relatively large pre-trained models (number of parameter is 90M for wav2vec2-base and 315M for wav2vec2-large and xlsr. It would be interesting to investigate if small pre-trained models can also be pruned and whether Observation 1 holds for them.

3. Our wav2vec2-base and wav2vec2-large are both pre-trained on Librispeech 960 hours. Another lack of study is the effect of pre-training data selections – what happens if pre-training and fine-tuning data are from different sources?

4. Our fine-tuning dataset (Librispeech and CommonVoice) are both read speech. Experiments on conversational (e.g. telephone) speech should be investigated.

5. In addition, though opposite to our motivation, it is unclear is the results hold for high-resource languages (e.g. 100h~1000h of fine-tuning data).

6. We presented the cross-lingual mask transfer experiments in H2L and CSR, as well as cross-task transfer for pre-trained BERT. It would be interesting to see if similar phenomena can be concluded for cross-task transfer for pre-trained wav2vec2.

7. Our ASR experiments are based on self-supervised pre-trained models. It remains to be studied on applying PARP to E2E ASR without self-supervised pre-training.

8. To isolate the effect of pruning, we did not conduct LM decoding/self-training for our pruned models. It is unclear if the same LM decoding/self-training used in prior art works for our pruned models.

9. Lastly, we note that this study is scientific by nature. Observation 1 emerges after our initial pilot study, and it motivates the central idea of PARP. We will leave it to follow-up work to test whether such pruning method is effective in more realistic settings (e.g. noisy data, limited bandwidth, etc).

---

15For example, we notice the recently published speech representation benchmark (similar to the GLUE benchmark but for speech), and an follow-up experiment would be to see if cross-task mask transfer holds across different downstream speech tasks: [https://github.com/s3prl/s3prl](https://github.com/s3prl/s3prl)