EH-MAM: Easy-to-Hard Masked Acoustic Modeling for Self-Supervised Speech Representation Learning

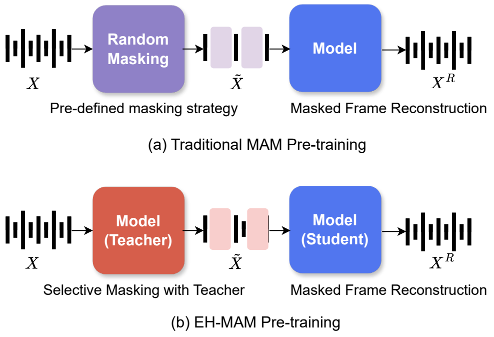
Anonymous ACL submission

Abstract

In this paper, we present EH-MAM (Easy-to-Hard adaptive Masked Acoustic Modeling), a novel self-supervised learning approach for speech representation learning. In contrast to the prior methods that use random masking schemes for Masked Acoustic Modeling (MAM), we introduce a novel selective and adaptive masking strategy. Specifically, during SSL training, we progressively introduce *harder* regions to the model for reconstruction. Our approach automatically selects hard regions and is built on the observation that the reconstruction loss of individual frames in MAM can provide natural signals to judge the difficulty of solving the MAM pre-text task for that frame. To identify these hard regions, we employ a teacher model that first predicts the frame-wise losses and then decides which frames to mask. By learning to create challenging problems, such as identifying harder frames and solving them simultaneously, the model is able to learn more effective representations and thereby acquire a more comprehensive understanding of the speech. Quantitatively, EH-MAM outperforms several state-of-the-art baselines across various low-resource speech recognition and SUPERB benchmarks by 5%-10%. Additionally, we conduct a thorough analysis to show that the regions masked by EH-MAM effectively capture useful context across speech frames.

1 Introduction

Self-supervised learning (SSL) has emerged as one of the most effective paradigms of speech representation learning when labeled data is scarce (Baevski and Mohamed, 2020; Mohamed et al., 2022). The task is to learn general-purpose speech representations from unlabeled data that can then be transferred to Spoken Language Processing (SLP) tasks like Automatic Speech Recognition (ASR), Speech Emotion Recognition (SER), etc (Huang et al., 2001). Progress in SSL for speech has led to significant performance improvements in a range of low-resource SLP tasks including Phoneme Recognition (PR), Keyword Spotting (KS), etc (Mohamed et al., 2022). Masked Acoustic Modeling (MAM) has been one the most prevalent pretext tasks for SSL-based speech representation learning wherein the model tries to reconstruct frames that are masked at the input, utilizing the context of the surrounding frames (Baevski et al., 2022, 2023; Liu et al., 2024).


Random Masking
Model
X
$\tilde{X}$
X^R
Pre-defined masking strategy
Masked Frame Reconstruction
Model (Teacher)
Model (Student)
Selective Masking with Teacher

(a) Traditional MAM Pre-training

(b) EH-MAM Pre-training

Figure 1: EH-MAM compared to random masking schemes employed widely in the literature. EH-MAM first identifies which frames to mask using a Teacher model and then solves the MAM task by reconstructing the selected masked regions using a Student model.

Although a considerable amount of research in MAM has been performed, most has focused on improving model architectures (Baevski et al., 2022; Chang et al., 2022; Baevski et al., 2023) and pretext tasks (Hsu et al., 2021; Lodagala et al., 2023; Liu et al., 2024), with very limited progress in improving the masking algorithm (Yue et al., 2022; Baevski et al., 2023). Most MAM algorithms still perform random masking of input frames. On the other hand, selective masking strategies for other domains, like computer vision (CV) (Bao et al., 2021; He et al., 2022; Kakogeorgiou et al., 2022) and natural language processing (NLP) (Sadeq

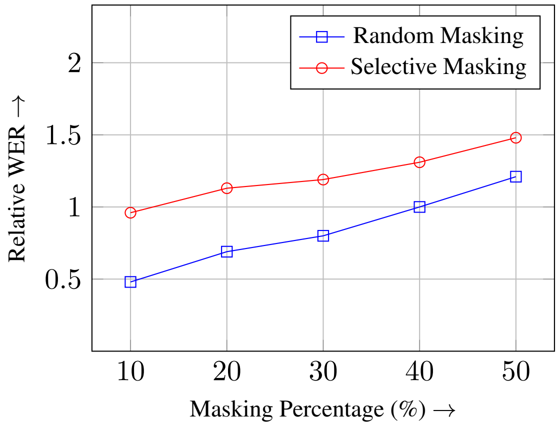


Figure 2: Increase in relative WER using selective and random masking schemes. During inference, under similar experimental settings, we selectively mask the frames with high reconstruction values and compare it against random masking. The former consistently shows a significant increase in relative WER than the later, thereby indicating that these frames capture more useful context for speech reconstruction as a result of capturing more information, Thus building on this result we hypothesize that asking a model to reconstruct these frames will result in stronger learning signals.

et al., 2022a, 2023; Xiao et al., 2022) that focuses on masking useful context, have shown significant improvements over random masking. This can be attributed to multiple factors, including: *(1) Variable Information Content:* Variable information content in data translates to variable learning signals for the reconstruction task. For instance, in Masked Language Modeling (MLM) (Devlin et al., 2019), the reconstruction of high-frequency stop words such as “the” or “is” offers minimal discriminative power due to the ubiquity and low semantic load of these words (Sadeq et al., 2022a, 2023). In speech, for example, this can be translated to reconstructing frames corresponding to random noise or partial phonemes, where much of the frames is already available as context. *(2) Progressive Learning:* Random masking fails to imitate the progressive human learning process (Madan et al., 2024). Humans do not receive knowledge uniformly; instead, they are exposed to progressively more complex information as they advance in the learning process. Mimicking this progression in the masking algorithm by initially exposing the model to simpler, more predictable speech patterns and gradually introducing more complex, less predictable ones can significantly enhance the learning trajectory. This approach aligns better with how humans learn, moving from simpler to more complex information, and helps the model develop a deeper understanding of language over time.

Main Contributions. To overcome the aforementioned problems, in this paper, we propose EH-MAM (**E**asy-To-**H**ard adaptive **M**asked **A**coustic **M**odelling), a novel selective and adaptive masking scheme for MAM. We build EH-MAM on the core hypothesis that *hard regions, characterized by collections of speech frames that are more difficult to reconstruct, serve as stronger signals for the learning process*. Fig. 2 shows the results of a simple experiment we performed to validate our hypothesis. By selectively masking *hard* regions, we notice a greater and consistent drop in WER performance for ASR. This suggests that masking hard regions captures useful context in the speech input. Our main contributions are as follows:

- We propose EH-MAM, a novel self-supervised speech representation learning algorithm. In contrast to solving a predefined MAM pre-text task, such as reconstructing randomly masked frames, EH-MAM aims to generate and align itself towards a more formidable MAM pre-text task. For generating a challenging MAM pre-text task, we first identify a collection of difficult frames to reconstruct, also called *hard regions*, followed by selectively masking them. We propose a lightweight *loss predictor* (introduced in Section 3.2.2) that predicts the frame-level reconstruction loss values and determines *hard regions* based on the output. To train the loss predictor jointly with MAM, we design a novel *auxiliary loss* (introduced in Section 3.2.2) that forces the predictor to learn the relative correlations between speech frames. Finally, to align the model towards reconstructing hard regions, we propose an *easy-to-hard* masking strategy (introduced in Section 3.2.3) that guides the overall EH-MAM learning process.

- We show the effectiveness of the speech representation learned by EH-MAM through extensive evaluations on low-resource speech recognition benchmarks (Kahn et al., 2020) and downstream evaluation on SUPERB (wen Yang et al., 2021). EH-MAM beats prior arts with a relative improvement of 5%-10%

- We perform a comprehensive analysis to demonstrate that regions masked by the EH-MAM effectively capture useful context across speech input.

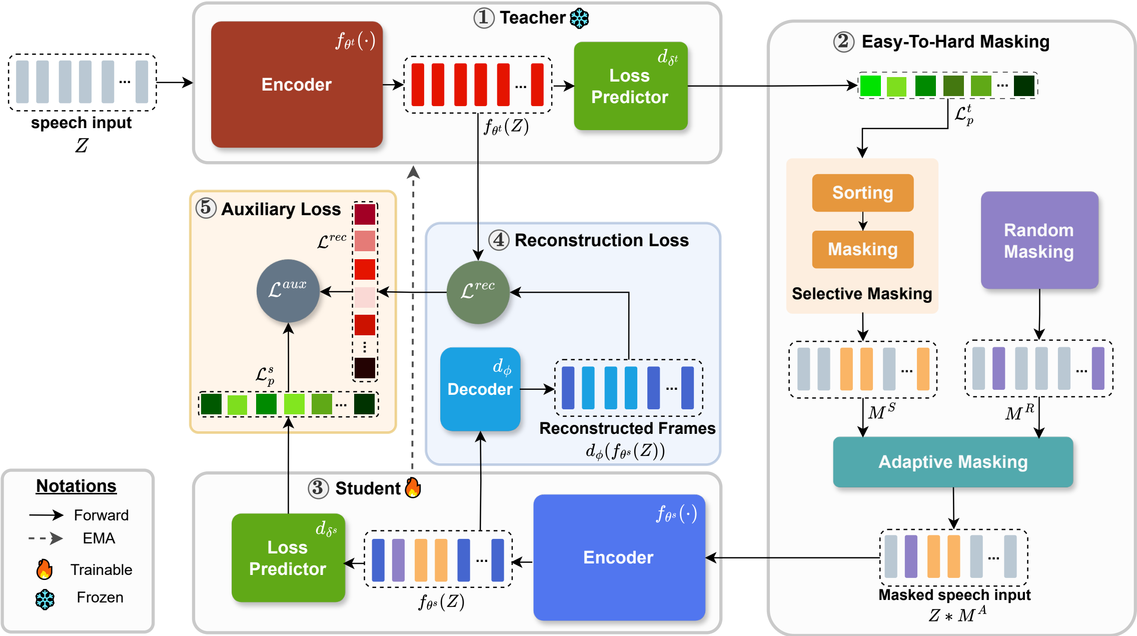


Figure 3: Illustration of EH-MAM SSL algorithm. EH-MAM employs the self-distillation SSL framework that consists of identical student and teacher networks. At each training iteration, the teacher is updated by the exponential moving average (EMA) of the student. ① For a speech input Z, we first use the teacher network to identify the speech frames that are hard to reconstruct, also called as hard regions. To achieve this, we predict the frame-level reconstruction loss values $\mathcal{L}_p^t$ using a loss predictor d_{δ^t} by feeding Z to the teacher network. ② Next, we utilize our *easy-to-hard* masking strategy to identify the mask indices M^S associated with hard regions, followed by progressively introducing them with random mask indices M^R over each epoch. ③ Finally, a masked variant $\tilde{Z}$ is fed to the student network, where it is tasked to ④ reconstruct masked regions by optimizing a reconstruction loss (as shown in Eqtn. 3) and ⑤ train a loss predictor d_{δ^s} by computing an auxiliary loss between predicted and original reconstruction loss values, $\mathcal{L}_p^s$ and $\mathcal{L}^{rec}$ respectively (as shown in Eqtn. 6).

2 Related Work

Self-Supervised Learning. SSL has emerged as a prevalent speech representation learning paradigm, demonstrating impressive downstream performance under low-resource settings (Lee et al., 2022; Mohamed et al., 2022). At its core, SSL relies on the quality of pretext tasks for capturing varied learning signals from unlabeled data sources. Based on the nature of the pretext tasks, the SSL frameworks are further categorized into the following sub-categories: 1) Contrastive Approaches: The pretext task is designed to maximize latent space similarity between the anchor and positive samples while minimizing the similarity between the anchor and negative samples. 2) Generative Approaches: These methods primarily focus on first building a target by randomly masking multiple speech frames and then reconstructing them by optimizing a similarity measure (MSE or Cross Entropy) between the predicted frames and the targets. The pretext task includes predicting future input from past inputs (Oord et al., 2018; Yang et al., 2022), masked from unmasked (Baevski et al., 2022, 2023) or the original from some other corrupted view (Lodagala et al., 2023). Masked Acoustic Modeling has undoubtedly seen the most success for speech representation learning.

Masked Acoustic Modeling (MAM) Conventional MAM architectures first perform frame-level masking, where randomly selected speech frames are masked using various existing masking strategies, including block or random masking (Bao et al., 2021; He et al., 2022). Next, they either employ a single encoder network like BERT (Devlin et al., 2019) to predict masked regions in a speech input (Liu et al., 2020; Chang et al., 2022; Chen et al., 2022; Hsu et al., 2021) or utilize self-distillation methods, where the student learns to reconstruct masked information under the guidance of an identical teacher network (Baevski et al., 2022, 2023; Liu et al., 2024). Although a considerable amount of research in MAM has undergone towards improving model architecture (Baevski et al., 2022, 2023) and introducing novel pretext tasks (Liu et al., 2024), developing better masking strategies is still under-explored.

3 Methodology

In this Section, we explain the EH-MAM methodology. We first provide an overview of the EH-MAM learning paradigm (in Section 3.1), followed by details on the reconstruction and auxiliary loss formulations (in Sections 3.2.1, 3.2.2). Finally, we introduce the *easy-to-hard* masking algorithm (in Section 3.2.3).

3.1 Overview of EH-MAM

We illustrate EH-MAM in Fig. 3. At its core, EH-MAM incorporates the *self-distillation* based SSL training paradigm for solving MAM pretext task, similar to Baevski et al. (2022, 2023). Specifically, EH-MAM consists of two identical networks, a teacher $\{f_{\theta^t}, d_{\delta^t}\}$ and a student $\{f_{\theta^s}, d_{\delta^s}\}$. A separate decoder d^R_ϕ is employed for reconstructing masked frames from the student representations. The context encoders f_θ are built using K-layered transformers (Vaswani et al., 2017), whereas the decoder d^R_ϕ and the loss predictor d_δ are constructed with light-weight D-layered 1D-convolution layers (Kiranyaz et al., 2019). During pre-training, the teacher parameters θ^t, δ^t are updated by performing exponential moving average (EMA) of the student parameters θ^s, δ^s (Tarvainen and Valpola, 2017). Formally, we define the update as follows:

$$\omega^t = \lambda\omega^t + (1-\lambda)\omega^s \tag{1}$$

where $\omega^t = \{\theta^t, \delta^t\}$, $\omega^s = \{\theta^s, \delta^s\}$, and λ is the decay rate. The student and decoder parameters are updated using gradient descent.

At each training iteration, we first extract low-frequency feature representations $Z \in \mathbb{R}^{N\times d}$ from raw speech signals $x \in X$ (Baevski et al., 2020) and feed it to the teacher network to get frame-level predicted reconstruction loss values $\mathcal{L}^t_p = d_{\delta^t}(f_{\theta^t}(Z))$. With the help of $\mathcal{L}^t_p$, we generate binary mask indexes $M^A = \{0,1\}^N$ using the *easy-to-hard* masking strategy (introduced in Section 3.2.3), followed by creating a masked version of the original speech input $\tilde{Z} \leftarrow Z \cdot M^A$. Finally, the student is trained with gradient descent to minimize a weighted combination of reconstruction loss $\mathcal{L}^{rec}$ (introduced in Section 3.2.1) and an auxiliary loss $\mathcal{L}^{aux}$ (introduced in Section 3.2.2). Formally, we define the objective function below:

$$\mathcal{L}^{joint} = \mathcal{L}^{rec} + \alpha\mathcal{L}^{aux} \tag{2}$$

where α is a balancing parameter and is set to 0.05 throughout the experiments (further ablation on this can be found in Appendix C.2).

Algorithm 1 Easy-To-Hard Masking

```
Require: train set $\mathcal{Z}$, masking probability $\mathcal{P}$, selective masking $M^S$, random masking $M^R$, adaptive masking $M^A$, number of frames $F(\cdot)$
Require: $t \in \{1, 2, ..., T\}$: training iteration
  for $t \leq T$ do
    Sample batch of training data $z \sim \mathcal{Z}$
    Compute selective and random masking probability: $\mathcal{P}^S \leftarrow \mathcal{P} \times \frac{t}{T}$, $\mathcal{P}^R \leftarrow 1 - \mathcal{P}^S$ respectively.
    Predict the reconstruction value $\mathcal{L}^t_p$ for each frame: $\mathcal{L}^t_p \leftarrow d^T_\delta(f_{\theta^t}(z))$
    Update $M^S$ by selecting frame indices corresponding to the top $k$ reconstruction values where $k = \lfloor P^S F(z) \rfloor$.
    Update $M^R$ by randomly selecting $\lfloor P^R F(z) \rfloor$ frame indices
    Update $M^A$ by taking the union of $M^S$ and $M^R$: $M^A \leftarrow M^S \cup M^R$
    Create a masked counterpart $\tilde{z} \leftarrow M^A \cdot z$
  end for
```

3.2 Selective Masking with EH-MAM

Motivation: EH-MAM distinguishes itself from the conventional self-distillation-based SSL training methods that are fixated on solving a predefined MAM task generated using random masking (Baevski et al., 2022; Chen et al., 2022; Chang et al., 2022), by enforcing the teacher to generate more challenging pretext tasks. To achieve this, EH-MAM first uses the teacher to identify hard regions, a collection of speech frames that are difficult to reconstruct, and then selectively mask these hard regions to create challenging MAM pretext tasks for the student to solve. Being constantly challenged by the teacher further directs the student to develop a much more nuanced understanding of speech. Additionally, we take inspiration from the recent studies in NLP and CV that have highlighted the significance of generating formidable pretext tasks for MLM (Masked Language Modeling) and MIM (Masked Image Modeling) using selective masking (Bao et al., 2021; Sadeq et al., 2022b).

To reweigh the model attention towards reconstructing such hard regions, we introduce the loss predictors d_{δ^s}, d_{δ^t} for the student and teacher networks, respectively. Further to train the loss predictor, we also propose an auxiliary objective function $\mathcal{L}^{aux}$, that the model optimizes alongside the reconstruction loss $\mathcal{L}^{rec}$.

3.2.1 Reconstruction Loss

As shown in Fig. 3, we first reconstruct the masked frames by feeding student representations $f_{\theta^s}(\tilde{Z})$ to a decoder d_ϕ^R. Similar to Baevski et al. (2022, 2023), the goal of d_ϕ^R is to reconstruct the teacher representation for time steps that are masked in the student input. To achieve this, we compute a reconstruction loss $\mathcal{L}^{rec}$ between the student and the teacher representations. Formally, we define reconstruction loss $\mathcal{L}^{rec}$ as follows:

$$\mathcal{L}^{rec} = \|M^A \cdot f_{\theta^t}(Z) - d_\phi^R(\cdot f_{\theta^s}(\tilde{Z}))\|_2^2 \qquad (3)$$

where $M^A \cdot f_{\theta^t}(Z)$ represents teacher representations associated with the masked speech input.

3.2.2 Loss Predictor and Auxiliary Loss

Motivation: Given the sequence of frame-level reconstruction loss values $\mathcal{L}^{rec} \in \mathbb{R}^N$, our goal is to create a challenging MAM pretext task for the student by selectively masking frames with high reconstruction values. As original reconstruction loss values $\mathcal{L}^{rec}$ are computed only for the masked regions (see Section 3.2.1), it provides limited information for deciding which frames to mask. To mitigate this problem, we introduce lightweight loss predictors $d_{\delta^s}, d_{\delta^t}$, which can be easily integrated with the student-teacher network, and add reconstruction loss predicting capabilities across both networks. To train these loss predictors, we propose a novel auxiliary loss $\mathcal{L}^{aux}$ that guides it towards capturing relative correlations between individual frames rather than forcing the predictor to generate exact frame-level reconstruction values.

Specifically, for each masked frame (i, j) where $i \neq j$ and $(i, j) \in \{1, 2, ..., N\}$, if $\mathcal{L}_i^{rec} > \mathcal{L}_j^{rec}$ than the predicted counterpart $\mathcal{L}_p^s = d_{\delta^s}(f_{\theta^s}(\tilde{Z}))$ must also have $\mathcal{L}_{p_i}^s > \mathcal{L}_{p_j}^s$. To formulate this constraint as a differentiable objective function, we first define a target distribution as an indicator variable I that captures the relative correlations between original reconstruction loss values, such as $\mathcal{L}_i^{rec} > \mathcal{L}_j^{rec}$. Formally we define this as follows:

$$I_{i,j} = \begin{cases} 1, & \mathcal{L}_i^{rec} > \mathcal{L}_j^{rec} \text{ and } \{i,j\} \in M^A \\ 0, & \text{otherwise} \end{cases} \qquad (4)$$

Next, similar to I, we introduce a predicted distribution S for representing the relative differences in the predicted reconstruction values $\mathcal{L}_p^s$. S is formally defined with a *sigmoid* function as:

$$S_{i,j} = e^{(\mathcal{L}_{p_i}^s - \mathcal{L}_{p_j}^s)}/(1 + e^{(\mathcal{L}_{p_i}^s - \mathcal{L}_{p_j}^s)}) \qquad (5)$$

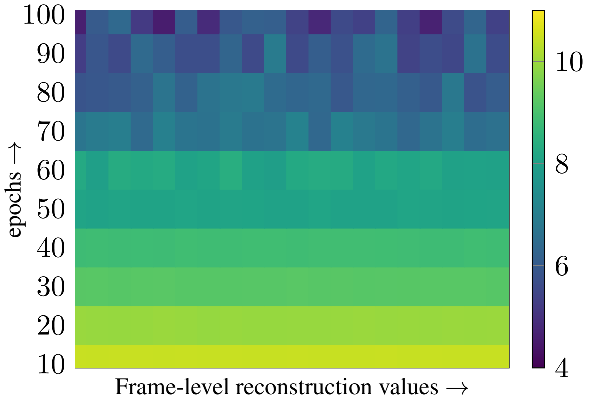


Figure 4: For a random speech utterance, we show the variation in frame-level reconstruction loss values across training epochs. During the initial stages of EH-MAM pre-training, we find that the model exhibits high frame-level reconstruction loss values, which results in low distinctiveness amongst individual values. This leads to increased stochasticity in the selective masking.

where $S_{i,j} > 0.5$ if $\mathcal{L}_{p_i}^s > \mathcal{L}_{p_j}^s$. Finally, we formulate our auxiliary objective function $\mathcal{L}^{aux}$ by first computing a vanilla cross entropy $\mathcal{H}(\cdot)$ between the target distribution I and the predicted distribution S: $\mathcal{L}^{aux} \leftarrow \mathcal{H}(I, S)$ and then minimizing it jointly with the reconstruction loss. We define the formulation of $\mathcal{L}^{aux}$ below:

$$\mathcal{L}^{aux} = -\sum_{i=1}^{N}\sum_{j=1}^{N} I_{i,j} \log S_{i,j} + \tilde{I}_{i,j} \log(1 - S_{i,j}) \qquad (6)$$

where $\tilde{I}_{i,j} = 1 - I_{i,j}$. $\{i, j\} \in M^A$ means that the i and j frames are masked during pre-training.

3.2.3 Selecting Hard Regions for Reconstruction

Motivation: Fig. 4 shows that during the initial stage of a EH-MAM pre-training, the reconstruction loss values are significantly high and exhibit low discriminative power ($L_i^{rec} \approx L_j^{rec}$). This leads to increased stochasticity in the overall selective masking process. Thus, inspired by the general human learning approach, where humans do not perceive knowledge uniformly but are subjected to a learning environment where they progressively comprehend more complex information, we propose an *easy-to-hard* masking strategy that guides the model to progressively mask *harder* regions for reconstruction. Specifically, we linearly increase the proportion of mask indices associated with hard regions at each training epoch. We define *hard* regions as a collection of speech frames that the model finds difficult to reconstruct.

We illustrate the masking strategy in Fig. 3. At each training iteration t and with a masking percentage P, we first compute P^S and P^R, the individual masking percentages for selective and random masking respectively. Precisely, we update P^S and P^R linearly as $P^S = P \times \frac{t}{T}$ and $P^R = 1 - P^S$, where T is the total number of training iterations. In selective masking, for each sampled batch $z \in Z$, we build mask M^S by selecting frame indices associated with the top k predicted reconstruction values $\mathcal{L}_p^t$. We use $k = \lfloor P^S F(z) \rfloor$, where $F(z)$ denotes the number of speech frames for an input batch z. To build a random mask M^R, we randomly sample $\lfloor P^R F(z) \rfloor$ frame indices. Finally, we compute the adaptive mask M^A by taking a union of M^S, M^R. We summarize the complete process of easy-to-hard masking in Algorithm 1

4 Experimental Setup

Pre-training Following Baevski et al. (2022, 2023); Liu et al. (2024), we pre-trained our model with 960 hours of unlabelled speech from LibriSpeech corpus (Panayotov et al., 2015). Due to resource constraints, we use a base variant of the context encoder (Baevski et al., 2020), with the number of transformer layers K = 12 and masking percentage = 50%. For the loss predictor and the reconstruction decoder, we utilize 1D-convolution layers, with the number of convolution layers D = 4. Moreover, a balancing parameter α is introduced and set to 0.05 during the joint optimization of reconstruction and auxiliary loss. All the pre-training experiments are performed on 4 × A100 40GB GPUs, for 400k updates and using a batch size of 63 minutes of speech (Additional details on the hyper-parameters used in EH-MAM can be found in Section C.3).

Fine-tuning Similar to Liu et al. (2024), to show the effectiveness of the learned speech representation, we fine-tune only the student counterpart with an additional CTC layer (Graves et al., 2006). We conduct a comprehensive evaluation under a low-resource labeled data setting using a 10 mins / 1 hour / 10 hours split from LibriLight benchmark (Kahn et al., 2020) and 100 hours split from Librispeech (Panayotov et al., 2015). For all the splits, we follow a similar fine-tuning setup as wav2vec2 (Baevski et al., 2020) (We provide additional fine-tuning details for all the splits in the Appendix B.1). We also perform a SUPERB (Speech Processing Universal PERformance Benchmark) evaluation (wen Yang et al., 2021), where a separate prediction head is trained on top of a frozen pre-trained model for various downstream tasks (Additional details on the downstream tasks present in SUPERB can be found in Appendix C.1)

Model	Content		Semantic			
	PR	ASR	KS	IC	SF	
	PER ↓	WER ↓	Acc ↑	Acc ↑	F1 ↑	CER ↓
wav2vec 2.0	5.47	6.43	96.23	92.35	88.30	24.77
HuBERT	5.41	6.42	96.30	98.34	88.53	25.20
WavLM	4.84	6.31	96.79	**98.63**	89.38	22.86
data2vec	4.69	4.94	96.56	97.63	88.59	25.27
DinoSR	**3.21**	**4.71**	96.89	98.02	88.83	23.57
data2vec 2.0	3.93	4.91	96.89	98.01	88.24	22.09
EH-MAM	3.86	4.89	**97.01**	98.01	**89.47**	**22.04**

Table 1: Results on Speech Processing Universal PERformance Benchmark (SUPERB). The downstream tasks include phoneme recognition (PR), automatic speech recognition (ASR), keyword spotting (KS), intent classification (IC), and slot filling (SF). The evaluation metrics used are accuracy (Acc), phoneme error rate (PER), word error rate (WER), f1 score (F1), and concept error rate (CER). The best and the second best results are **bolded** and underlined respectively.

Baselines We compare the performance of EH-MAM across various SSL-based speech representation learning baselines that employ 1) single encoder: wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021) and 2) self-distillation network: data2vec (Baevski et al., 2022), data2vec 2.0 (Baevski et al., 2023) and DinoSR (Liu et al., 2024) to reconstruct masked frames. All the baselines share similar BASE encoder configurations as mentioned in Table 5. Due to compute constraints, we avoid retraining the baselines from scratch and use the checkpoints open-sourced by the authors.

Dataset and Evaluation Metric We pre-train EH-MAM on 960 hours of unlabelled speech data from the LibriSpeech corpus (Panayotov et al., 2015). Further, we evaluate EH-MAM on a wide range of speech-related downstream tasks, including 1) Low resource ASR benchmarks: Libri-Light (Kahn et al., 2020), 100 hours LibriSpeech corpus (Panayotov et al., 2015), Wall-Street Journal (WSJ) (Paul and Baker, 1992), SwitchBoard (Godfrey et al., 1992) and 2) SUPERB evaluation: a collection of a diverse set of downstream tasks including Phoneme Recognition (PR), Automatic Speech Recognition (ASR), Keyword Spotting (KS), Intent Classification (IC) and Slot Filling (SF). Additional details on duration, train/test splits, and evaluation metrics can be found in Table 4.

Models	Pre-training steps	Batch size (minutes)	dev (WER ↓)		test (WER ↓)	
			clean	other	clean	other
10 minutes labeled data						
wav2vec 2.0 (Baevski et al., 2020)	400k	96	8.9	15.7	9.1	15.6
HuBERT (Hsu et al., 2021)	250k + 400k	47	9.1	15.0	9.7	15.3
data2vec (Baevski et al., 2022)	400k	63	7.3	11.6	7.9	12.3
DinoSR (Liu et al., 2024)	400k	63	6.6	10.8	7.3	11.8
data2vec 2.0 (Baevski et al., 2023)	400k	17	6.4	10.5	7.2	11.5
Eh-MAM	400k	63	**6.3**	**10.2**	**7.1**	**11.1**
1 hr labeled data						
wav2vec 2.0 (Baevski et al., 2020)	400k	96	5.0	10.8	5.5	11.3
HuBERT (Hsu et al., 2021)	250k + 400k	47	5.6	10.9	6.1	11.3
WavLM (Chen et al., 2022)	250k + 400k	187	-	-	5.7	10.8
data2vec (Baevski et al., 2022)	400k	63	4.0	8.5	4.6	9.1
DinoSR (Liu et al., 2024)	400k	63	4.1	8.1	4.6	8.7
data2vec 2.0 (Baevski et al., 2023)	400k	17	4.0	8.0	4.6	8.7
Eh-MAM	400k	63	**4.0**	**7.8**	**4.6**	**8.7**
10 hr labeled data						
wav2vec 2.0 (Baevski et al., 2020)	400k	96	3.8	9.1	4.3	9.5
HuBERT (Hsu et al., 2021)	250k + 400k	47	3.9	-	4.3	9.4
WavLM (Chen et al., 2022)	250k + 400k	187	-	-	4.3	9.2
data2vec (Baevski et al., 2022)	400k	63	3.3	7.5	3.9	8.1
DinoSR (Liu et al., 2024)	400k	63	3.1	7.0	3.6	7.6
data2vec 2.0 (Baevski et al., 2023)	400k	17	3.0	7.0	3.4	7.6
Eh-MAM	400k	63	**3.0**	**6.8**	**3.3**	**7.3**
100 hr labeled data						
wav2vec 2.0 (Baevski et al., 2020)	400k	96	2.7	7.9	3.4	8.0
HuBERT (Hsu et al., 2021)	250k + 400k	47	2.7	7.8	3.4	8.1
WavLM (Chen et al., 2022)	250k + 400k	187	-	-	3.4	7.7
data2vec (Baevski et al., 2022)	400k	63	2.2	6.4	2.8	6.8
DinoSR (Liu et al., 2024)	400k	63	2.3	6.4	2.9	6.7
data2vec 2.0 (Baevski et al., 2023)	400k	17	2.2	6.2	2.8	6.4
Eh-MAM	400k	63	**2.2**	**6.1**	**2.8**	**6.3**

Table 2: Results on LibriLight benchmark and LibriSpeech for ASR. All the models share a similar BASE size encoder and are first fine-tuned with a 10 min / 1hr / 10hr / 100hr labeled dataset and then evaluated on common dev/test splits. The evaluation metric used is word error rate (WER). The best and the second best results are **bolded** and underlined respectively

5 Results and Analysis

In this section, we present the quantitative and qualitative results. For quantitative evaluation, we first fine-tune Eh-MAM on LibriLight (Kahn et al., 2020) and evaluate across all the test splits. Next, to show the scalability of the speech representations learned by Eh-MAM, we conduct a downstream evaluation on SUPERB benchmark (wen Yang et al., 2021). Additionally, we also perform a qualitative analysis on the masked regions predicted by the Eh-MAM. All the results reported for Eh-MAM are averaged across five runs.

5.1 Evaluation on Low-Resource ASR

For low-resource ASR evaluation, we follow a similar procedure as Baevski et al. (2020) wherein we fine-tune only the student counterpart of Eh-MAM with an additional CTC layer (Graves et al., 2006) on top. We perform fine-tuning using low-resource labeled datasets under four different setups, 10min / 1hour / 10hour from LibriLight (Kahn et al., 2020) and 100hour Librispeech (Panayotov et al., 2015). For evaluation, we use the standard dev/test split of Librispeech and report the word error rate (WER) by decoding with the official 4-gram language model. Following the prior work Baevski et al. (2022, 2023), the decoding hyper-parameter is searched with Ax (refer to Section C.1). As shown in Table 2, Eh-MAM consistently outperforms all the prior SSL methods across all the setups. We also provide additional results on other low-resource ASR benchmarks such as Wall Street Journal (WSJ) (Paul and Baker, 1992) and Switch-Board (SB) (Godfrey et al., 1992) in Appendix E.

5.2 Downstream Evaluation on SUPERB

We extensively evaluate the effectiveness and scalability of the speech representation learned by Eh-MAM using the Speech Processing Universal PERformance Benchmark (SUPERB). SUPERB, in total, consists of ten speech-related downstream tasks that aim to study four aspects of speech: con-

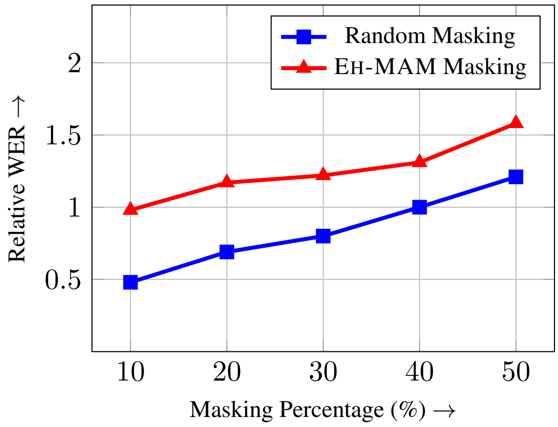


Figure 5: We compare the increase in relative Word Error Rate (WER) by selectively masking hard regions predicted by the loss predictor (EH-MAM Masking) Vs randomly masking frames. The increase in relative WER indicates that the EH-MAM Masking scheme masks useful context in an input.

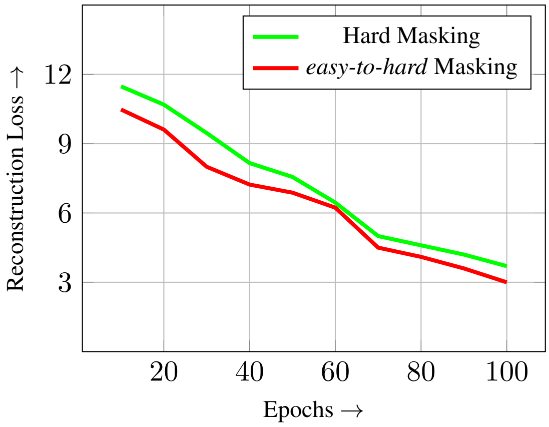


Figure 6: We compare the effectiveness of EH-MAM in optimizing a MAM pretext task, such as the reduction in reconstruction loss, with hard and *easy-to-hard* masking schemes. The *easy-to-hard* masking scheme shows better convergence in reconstruction loss compared to hard masking strategies.

tent, speaker, semantics, and paralinguistics. To investigate the model's capabilities to understand speech content and semantics, we report the results on phoneme recognition (PR), automatic speech recognition (ASR), keyword spotting (KS), intent classification (IC), and slot filling (SF) (Additional details on all the downstream tasks can be found in Appendix B.2). For downstream evaluation on SUPERB, we follow a similar setup as wen Yang et al. (2021), where we train a prediction head on top of the frozen pre-trained models instead of complete fine-tuning. As shown in Table 1, for semantic tasks like IC and SF, the EH-MAM outperforms prior art, showing its capabilities to capture better semantic information from speech input. On context tasks, EH-MAM surpasses prior art in KS and achieves comparable performance on PR and ASR.

5.3 Qualitative Analysis

EH-MAM mask useful context: To show EH-MAM does mask useful context, we conduct a simple experiment wherein during ASR inference, we selectively mask the frames with high predicted reconstruction value using the loss predictor and compare the increase in relative WER with random masking. As shown in Fig. 5, under SUPERB evaluation setting for ASR (refer Section 4), we find selectively masking frames with EH-MAM constantly shows a higher relative WER when compared to random masking across various masking percentages. Higher relative WER indicates that a selective masking scheme with the EH-MAM masks useful context in a speech input.

EH-MAM adapts well towards reconstructing hard regions: To show how well EH-MAM adapts towards reconstructing hard regions, we conduct an experiment wherein we compare the EH-MAM ability to reconstruct hard regions (collection of frames with high reconstruction values) using 1) *hard masking*, masking only hard regions at each epoch and 2) *easy-to-hard* masking, where we progressively introduce hard regions with randomly masked regions at each epoch. As shown in Fig. 6, while pre-training EH-MAM, *easy-to-hard* masking scheme shows better convergence in reconstruction loss when compared with hard masking strategies. This indicates that progressively introducing hard regions in an easy-to-hard manner, improves EH-MAM adaptability toward reconstructing masked regions during pre-training.

6 Conclusion

In this paper, we propose EH-MAM, a novel SSL framework for learning robust speech representations. In contrast to prior work that relies on random masking schemes for creating MAM pretext tasks, EH-MAM first identifies hard regions to reconstruct using a teacher network and then challenges the student to reconstruct them by progressively introducing hard regions throughout the learning process. Next, we introduce an *easy-to-hard* masking scheme that guides the EH-MAM to mask harder regions to reconstruct step-by-step. EH-MAM outperforms all the other models on popular low-resource ASR benchmarks and downstream evaluation on SUPERB.

7 Limitations and Future Work

EH-MAM and our experimental setup have a few limitations, as mentioned below:

- We do not employ a LARGE size encoder in EH-MAM, for example, a 24-layer variant used by Baevski et al. (2023) due to compute constraints.
- The loss-predictors used in EH-MAM increase the trainable parameter count compared to other baselines such as data2vec 2.0 (Baevski et al., 2023) during pre-training. However, we acknowledge that this accounts only for a slight increase in the total parameter count (roughly 5%).
- Due to recourse constraints, we conduct the downstream evaluation on SUPERB for context and semantic-related tasks. We plan to extend the evaluation across speaker and paralinguistic tasks in the future.

References

Alexei Baevski, Arun Babu, Wei-Ning Hsu, and Michael Auli. 2023. Efficient self-supervised learning with contextualized target representations for vision, speech and language. In *International Conference on Machine Learning*, pages 1416–1429. PMLR.

Alexei Baevski, Wei-Ning Hsu, Qiantong Xu, Arun Babu, Jiatao Gu, and Michael Auli. 2022. Data2vec: A general framework for self-supervised learning in speech, vision and language. In *International Conference on Machine Learning*, pages 1298–1312. PMLR.

Alexei Baevski and Abdelrahman Mohamed. 2020. Effectiveness of self-supervised pre-training for asr. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7694–7698. IEEE.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Heng-Jui Chang, Shu-wen Yang, and Hung-yi Lee. 2022. Distilhubert: Speech representation learning by layer-wise distillation of hidden-unit bert. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7087–7091. IEEE.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Zewen Chi, Shaohan Huang, Li Dong, Shuming Ma, Bo Zheng, Saksham Singhal, Payal Bajaj, Xia Song, Xian-Ling Mao, Heyan Huang, et al. 2021. Xlm-e: Cross-lingual language model pre-training via electra. *arXiv preprint arXiv:2106.16138*.

Alice Coucke, Alaa Saade, Adrien Ball, Théodore Bluche, Alexandre Caulier, David Leroy, Clément Doumouro, Thibault Gisselbrecht, Francesco Caltagirone, Thibaut Lavril, Maël Primet, and Joseph Dureau. 2018. Snips voice platform: an embedded spoken language understanding system for private-by-design voice interfaces. *Preprint*, arXiv:1805.10190.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

John J Godfrey, Edward C Holliman, and Jane McDaniel. 1992. Switchboard: Telephone speech corpus for research and development. In *Acoustics, speech, and signal processing, ieee international conference on*, volume 1, pages 517–520. IEEE Computer Society.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Xuedong Huang, Alex Acero, Hsiao-Wuen Hon, and Raj Reddy. 2001. *Spoken language processing: A guide to theory, algorithm, and system development*. Prentice hall PTR.

Jacob Kahn, Morgane Riviere, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. IEEE.

Ioannis Kakogeorgiou, Spyros Gidaris, Bill Psomas, Yannis Avrithis, Andrei Bursuc, Konstantinos Karantzalos, and Nikos Komodakis. 2022. What to hide from your students: Attention-guided masked image modeling. In *European Conference on Computer Vision*, pages 300–318. Springer.

Serkan Kiranyaz, Turker Ince, Osama Abdeljaber, Onur Avci, and Moncef Gabbouj. 2019. 1-d convolutional neural networks for signal processing applications. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8360–8364. IEEE.

Hung-yi Lee, Abdelrahman Mohamed, Shinji Watanabe, Tara Sainath, Karen Livescu, Shang-Wen Li, Shu-wen Yang, and Katrin Kirchhoff. 2022. Self-supervised representation learning for speech processing. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Tutorial Abstracts*, pages 8–13. Association for Computational Linguistics.

Alexander H Liu, Heng-Jui Chang, Michael Auli, Wei-Ning Hsu, and Jim Glass. 2024. Dinosr: Self-distillation and online clustering for self-supervised speech representation learning. *Advances in Neural Information Processing Systems*, 36.

Andy T. Liu, Shu-wen Yang, Po-Han Chi, Po-chun Hsu, and Hung-yi Lee. 2020. 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.

Vasista Sai Lodagala, Sreyan Ghosh, and Srinivasan Umesh. 2023. data2vec-aqc: Search for the right teaching assistant in the teacher-student training setup. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech model pre-training for end-to-end spoken language understanding. *Preprint*, arXiv:1904.03670.

Neelu Madan, Nicolae-Cătălin Ristea, Kamal Nasrollahi, Thomas B Moeslund, and Radu Tudor Ionescu. 2024. Cl-mae: Curriculum-learned masked autoencoders. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2492–2502.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, et al. 2022. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. 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 *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Douglas B Paul and Janet Baker. 1992. The design for the wall street journal-based csr corpus. In *Speech and Natural Language: Proceedings of a Workshop Held at Harriman, New York, February 23-26, 1992*.

Nafis Sadeq, Byungkyu Kang, Prarit Lamba, and Julian McAuley. 2023. Unsupervised improvement of factual knowledge in language models. *arXiv preprint arXiv:2304.01597*.

Nafis Sadeq, Canwen Xu, and Julian McAuley. 2022a. Informask: Unsupervised informative masking for language model pretraining. *arXiv preprint arXiv:2210.11771*.

Nafis Sadeq, Canwen Xu, and Julian McAuley. 2022b. Informask: Unsupervised informative masking for language model pretraining. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5866–5878, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems*, pages 1195–1204.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *Preprint*, arXiv:1804.03209.

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, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung yi Lee. 2021. Superb: Speech processing universal performance benchmark. *Preprint*, arXiv:2105.01051.

Shitao Xiao, Zheng Liu, Yingxia Shao, and Zhao Cao. 2022. Retromae: Pre-training retrieval-oriented language models via masked auto-encoder. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 538–548. Association for Computational Linguistics.

Gene-Ping Yang, Sung-Lin Yeh, Yu-An Chung, James Glass, and Hao Tang. 2022. Autoregressive predictive coding: A comprehensive study. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1380–1390.

Xianghu Yue, Jingru Lin, Fabian Ritter Gutierrez, and Haizhou Li. 2022. Self-supervised learning with segmental masking for speech representation. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1367–1379.

A Baseline Details

wav2vec 2.0. [1] (Baevski et al., 2020) The wav2vec 2.0 model integrates contrastive learning with masking. Similar to the CPC model (Oord et al., 2018), it employs the InfoNCE loss (Baevski et al., 2020) to maximize the similarity between a contextualized representation (anchors) and a localized representation (positives) simultaneously minimizing the similarity with other masked regions (negatives). Instead of directly using the contextualized representations, wav2vec 2.0 employs a separate quantization module to generate positives and negatives.

HuBERT. [2] (Hsu et al., 2021) Like BERT (Devlin et al., 2019), HuBERT follows a generative approach by discretizing the continuous MFCC features using the K-means algorithm and creating targets by randomly masking the quantized units. Unlike BERT, HuBERT employs a two-iteration training process wherein, in the first iteration, the model is trained to predict targets generated from the MFCC features, followed by quantizing the learned representations obtained from the first iteration training using K-means to generate new targets, which the model utilize in the second iteration training.

WavLM. [3] (Chen et al., 2022) WavLM extends the HuBERT's learning paradigm by introducing a gated relative position bias (Chi et al., 2021) at each transformer layer. Further, WavLM proposes an utterance-mixing strategy wherein training samples are augmented by mixing utterances from different speakers, and the targets are created from the original sample.

data2vec. [4] (Baevski et al., 2022) data2vec introduces a self-distillation-based student-teacher networks () for speech representation learning. The core idea is to predict the latent representations of the whole speech unlabeled data from the masked view. data2vec trains a student network by feeding a masked version of input to predict the latent representation obtained by feeding the whole input to a teacher network. The teacher's parameters are updated by the exponential moving average (ema) of the student's parameters.

data2vec 2.0. [5] (Baevski et al., 2023) data2vec 2.0 uses an identical learning objective as data2vec but with two key changes. Firstly, data2vec 2.0 introduces a lightweight decoder module that reconstructs the masked frames in student representation before maximizing the similarity with the teacher representations. Next, data2vec 2.0 employs a multi-mask strategy where multiple mask variants of the same input are fed to the student network, followed by calculating reconstruction loss for all the variants with a common teacher representation obtained from the original speech input.

DinoSR. [6] (Liu et al., 2024) DinoSR uses similar architecture as (Baevski et al., 2022), but introduces a novel gradient-free online clustering method for learning discrete acoustic units. DinoSR initially employs a teacher network to extract contextualized embeddings from the input audio. It then applies an online clustering scheme to these embeddings to create a machine-discovered phone inventory. Finally, it uses the discretized tokens to guide a student network.

B Dataset Details

B.1 ASR Evaluation

LibriSpeech. [10] (Panayotov et al., 2015) The LibriSpeech dataset is a widely-used corpus of English read speech with approximately 1000 hours of audiobooks available in the public domain, which includes a broad range of speakers, both male and

[1] https://github.com/facebookresearch/fairseq/tree/main/examples/wav2vec

[2] https://github.com/facebookresearch/fairseq/tree/main/examples/hubert

[3] https://huggingface.co/docs/transformers/en/model_doc/wavlm

[4] https://github.com/facebookresearch/fairseq/tree/main/examples/data2vec

[5] https://github.com/facebookresearch/fairseq/tree/main/examples/data2vec

[6] https://github.com/Alexander-H-Liu/dinosr

[10] https://www.openslr.org/12r

Dataset	Language	Domain	Type	Duration (hour) (train, dev, test)
LibriSpeech (Panayotov et al., 2015)	English	General	Read	960, 10, 10
Libri-Light (Kahn et al., 2020)	English	General	Read	11.16, 10, 10
SwitchBoard (SWBD) (Godfrey et al., 1992)	English	Call Cent.	Conv.	30, 5, N.A.
Wall Street Journal (WSJ) (Paul and Baker, 1992)	English	Finance	Read	80, 1.1, 0.4

Table 3: Detailed Statistics of datasets used in our low resource ASR evaluation. Type refers to Conversational or Read speech.

Task	Category	Dataset	Duration(hour) (train, dev, test)	Evaluation Metric
Phoneme Recognition(PR)	Content	LibriSpeech(Panayotov et al., 2015)	100, 5.4, 5.4	Phoneme Error Rate(PER)
Automatic Speech Recognition(ASR)	Content	LibriSpeech(Panayotov et al., 2015)	100, 5.4, 5.4	Word Error Rate(WER)
Keyword Spotting(KS)	Content	Speech Commands v0.1 [7](Warden, 2018)	18, 2, 1	Accuracy(Acc)
Intent Classification(IC)	Speaker	Fluent Speech Commands [8](Lugosch et al., 2019)	23.1, 3.2, 3.9	Accuracy(Acc)
Slot Filling(SF)	Speaker	Audio SNIPS [9](Coucke et al., 2018)	166.0, 9.0, 9.0	Concept Error Rate(CER)

Table 4: Details on downstream tasks and datasets used for the SUPERB evaluation

female, with diverse accents and ages, providing a rich source for speech and language research. We pre-train our model on 960 hours LibriSpeech unlabeled data and fine-tune for ASR evaluation on 100 hours of labeled data.

Libri-Light. [11] (Kahn et al., 2020) The Libri-light is a dataset derived from the LibriVox project, consisting of audiobooks in the public domain, much like the LibriSpeech dataset, but aims to address the limitations of traditional ASR datasets by providing 60 hours of unlabelled speech complemented with a smaller amount of labeled data. We conduct evaluation on ASR with labeled data of 10 mins / 1 hour / 10 hours split from LibriLight.

WSJ. [12] (Paul and Baker, 1992) The WSJ dataset consists of approximately 80 hours of read speech derived from articles in the Wall Street Journal, offering high-quality audio and transcriptions ideal for training and evaluating ASR systems. The WSJ dataset includes recordings from 84 speakers, providing diverse voice samples, including accurate word-level transcriptions for all audio files and metadata for speaker identities and recording conditions. We use the WSJ dataset for our ASR task evaluation on 80 hours of unlabeled data for training and 1.5 hours for of labeled data for testing.

Switchboard. [13] (Godfrey et al., 1992) The Switchboard is a telephone speech corpus consisting of approximately 260 hours of speech, which includes 2,400 two-sided telephone conversations among 543 speakers. The dataset conversations cover 70 different topics, from current events to personal interests, providing varied and natural discourse, making it an invaluable resource in the field of speech recognition, dialogue systems, and conversational analysis. We report the our ASR evaluation on 30 hours of unlabeled data for training and 5 hours of data for testing.

B.2 SUPERB (Speech processing Universal PERformance Benchmark)

SUPERB (wen Yang et al., 2021) is a leaderboard to benchmark the performance of a shared model across a wide range of speech processing tasks with minimal architecture changes and labeled data. The key focus here is to extract the representation learned from SSL and to learn task-specialized lightweight prediction heads on top of the frozen shared models. Below we detail the tasks in SUPERB that we use for evaluation.

Phoneme Recognition (PR) converts spoken language into its smallest units of sound, known as phonemes. This task incorporates alignment modeling to circumvent issues with incorrect forced alignments. The LibriSpeech (Panayotov et al., 2015) subsets train-clean-100/dev-clean/test-clean are utilized for training, validation, and testing in the SUPERB framework. The primary metric for evaluation is the phone error rate (PER).

[11] https://github.com/facebookresearch/libri-light

[12] https://catalog.ldc.upenn.edu/LDC93S6A

[13] https://catalog.ldc.upenn.edu/LDC97S62

[7] https://www.tensorflow.org/datasets/catalog/speech_commands

[8] https://fluent.ai/

[9] https://github.com/aws-samples/aws-lex-noisy-spoken-language-understanding

	Base (Librispeech)
GPUs	4
Learning rate	7.5×10^{-4}
Adam β_1 / β_2	0.9 / 0.98
Weight decay	0.01
Clip norm	-
Learning rate schedule	cosine
Warmup updates	8,000
Batch size	63 min
τ_0 (EMA start)	0.999
τ_e (EMA end)	0.99999
τ_n (EMA anneal steps)	75,000
B (block width)	5
R (mask ratio)	0.5
A (mask adjust)	0.05
K (layers to average)	8
Target normalization	IN → AVG
Updates	400,000
Decoder dim.	384
Decoder conv. groups	16
Decoder kernel	7
Decoder layers (D)	4
Loss Predictor dim.	384
Loss Predictor conv. groups	16
Loss Predictor kernel	7
Loss Predictor layers (D)	4

	10 Minutes	1 Hour	10 Hours	100 Hours
GPU	4	4	4	4
Learning rate	5.0×10^{-5}	5.0×10^{-5}	5.0×10^{-5}	3.0×10^{-5}
Adam β_1 / β_2	0.9/0.98	0.9/0.98	0.9/0.98	0.9/0.98
Learning rate schedule	tri_stage	tri_stage	tri_stage	tri_stage
Batch Size	32	32	32	32
Updates	13000	13000	20000	80000
Apply mask	true	true	true	true
Mask prob	0.65	0.65	0.65	0.65
Mask channel prob	0.25	0.25	0.50	0.50
Mask channel length	64	64	64	64
Layerdrop	0.1	0.1	0.05	0.1
Activation dropout	0.1	0.1	0.1	0.1
Freeze finetune updates	10000	10000	10000	0

Table 5: **(Left)** EH-MAM pre-training hyper-parameters. IN is instance normalization; AVG is mean pooling. **(Right)** EH-MAM fine-tuning hyper-parameters for LibriLight (Kahn et al., 2020)

Automatic Speech Recognition (ASR) transcribes spoken words into text. While PR focuses on the precision of phoneme modeling, ASR assesses improvements in terms of their practical relevance. The training, validation, and testing phases use the LibriSpeech (Kahn et al., 2020) subsets train-clean-100/dev-clean/test-clean. The word error rate (WER) serves as the evaluation metric.

Keyword Spotting (KS) involves the detection of specified keywords within speech, categorizing utterances into a set list of terms. The Speech Commands dataset v1.0 (Warden, 2018), which includes ten keyword categories, a silence category, and an "unknown" category for erroneous detections, is used in this task. Accuracy (ACC) is the metric for assessing performance.

Intent Classification (IC) assigns categories to spoken utterances to ascertain the speaker's intent. It employs the Fluent Speech Commands (Lugosch et al., 2019) dataset, where utterances are labeled according to three intent categories: action, object, and location. The evaluation metric here is also accuracy (ACC).

Slot Filling (SF) entails predicting a series of semantic slots from speech. The Audio SNIPS (Coucke et al., 2018) dataset, which features synthesized multi-speaker utterances for the SNIPS NLU benchmark, is used for this purpose. Evaluation is based on the slot-type F1 score and slot-value character error rate (CER).

C Additional Details: Hyper-Parameter Tuning

C.1 Pre-training and Fine-tuning

Table 5 summarize the hyper-parameter choices for EH-MAM when pre-training on Librispeech-960 hours (Panayotov et al., 2015) and fine-tuning across various LibriLight (Kahn et al., 2020) setups (10min / 1hour / 10hour). Most hyper-parameters are taken from the prior art (Baevski et al., 2023, 2022). For decoding, the hyper-parameter is searched using Ax [14]

α	1	0.5	0.1	0.05	0.01
WER ↓	7.4	7.3	7.3	**7.1**	7.2

Table 6: α = 0.05 gives the best performance

C.2 Balancing Parameter α

In Table 6 we show the effect of changing the balancing parameter α, on the final low-resource asr evaluation. Specifically, we pre-train EH-MAM

[14] https://github.com/facebook/Ax

with different balancing parameters and then perform an end-to-end fine-tuning on 10 mins setup of LibriLight (Kahn et al., 2020). Finally, we compute WER for the test-clean split.

C.3 Masking Probability $\mathcal{P}$

In Table 7 we show the effect of changing the masking probability $\mathcal{P}$, on the final low-resource asr evaluation. Specifically, we pre-train Eh-MAM with different masking probability and then perform an end-to-end fine-tuning on 10 mins setup of LibriLight (Kahn et al., 2020). Finally, we compute WER for the test-clean split.

$\mathcal{P}$	0.1	0.2	0.3	0.4	0.5
WER ↓	9.4	8.1	7.9	7.3	**7.1**

Table 7: $\mathcal{P}$ = 0.5 gives the best performance

Models	WSJ (WER ↓)		Switchboard (WER ↓)
	dev	test	dev
data2vec 2.0	9.2	9.0	15.2
Eh-MAM	**8.4**	**8.2**	**14.3**

Table 8: Performance comparison of Eh-MAM on WSJ and Switchnoard datasets.

D Additional Details: General

Compute details. For all our pre-training and fine-tuning experiments, we used four NVIDIA A100-40GB GPUs. The pre-training Eh-MAM requires five days of training and consists of 94.40M parameters. All the fine-tuning experiments on LibriLight (Kahn et al., 2020) require two days each. Additionally, individual downstream evaluation on SUPERB requires one day.

Potential Risk. As the Eh-MAM follows a self-supervised training regime, it may learn spurious correlations which can affect downstream performance on ASR, PR, etc. Moreover, Eh-MAM might get biased towards a particular type of accent, dialect, or domain, such as telephonic or read speech, due to a huge amount of unlabeled data, which may not be diverse.

Software and Packages details. We implement all our models in PyTorch [15] and use Fairseq [16] toolkit and SUPERB [17] for all our experiments.

E Additional Results

We present additional results for low-resource ASR evaluation on WSJ (Paul and Baker, 1992) and Switchboard (Godfrey et al., 1992). The evaluation settings for both datasets are similar to LibriLight (Kahn et al., 2020). Train/Test splits for both datasets can be found in Table 3. As shown in Table 8, Eh-MAM outperforms the state-of-the-art model, data2vec 2.0, across all the dev/test splits.

[15] https://pytorch.org/

[16] https://github.com/facebookresearch/fairseq

[17] https://superbbenchmark.github.io

Algorithm 2 Pseudo-Code of Easy-to-Hard Masking.

```
def compute_mask_indices_ema_loss(
    shape: Tuple[int, int],
    padding_mask: Optional[torch.Tensor],
    loss_pred: Optional[torch.Tensor],
    mask_prob: float,
    mask_length: int,
    current_epoch: int,
    total_epoch: int,
    mask_type: str = "static",
    min_masks: int = 0,
    require_same_masks: bool = True,
    mask_dropout: float = 0.0):

    bsz, all_sz = shape
    mask = np.full((bsz, all_sz), False)
    # add a random number for probabilistic rounding
    all_num_mask = int( mask_prob * all_sz / float(mask_length) + np.random.rand())
    # Get the loss lattice from decoder
    ids_shuffle_loss = torch.argsort(loss_pred, dim=1).cpu().detach().numpy()

    all_num_mask = max(min_masks, all_num_mask)
    # guide the making wrt to training epoch
    #keep_ratio = 1.0
    keep_ratio = float((current_epoch + 1) / total_epoch)
    mask_idcs = []
    for i in range(bsz):
        if padding_mask is not None:
            sz = all_sz - padding_mask[i].long().sum().item()
            num_mask = int(
                # add a random number for probabilistic rounding
                mask_prob * sz / float(mask_length)
                + np.random.rand()
            )
            num_mask = max(min_masks, num_mask)
        else:
            sz = all_sz
            num_mask = all_num_mask

        if mask_type == "static":
            lengths = np.full(num_mask, mask_length)
        else:
            raise Exception("unknown␣mask␣selection␣" + mask_type)

        if sum(lengths) == 0:
            lengths[0] = min(mask_length, sz - 1)

        min_len = min(lengths)
        if sz - min_len <= num_mask:
            min_len = sz - num_mask - 1
        #reverse the index list to get the indexes associated with max losses
        sample_loss_index = ids_shuffle_loss[i][::-1]

        #calculate random_mask and learnable_mask using keep_ratio
        random_mask = int(num_mask * (1-keep_ratio))
        learnable_mask = num_mask - random_mask

        #randomly select mask index.
        mask_idc = np.random.choice(sz - min_len, num_mask, replace=False)
        sample_loss_index = sample_loss_index[:learnable_mask]

        #recalculate mask_idc for random masking:
        mask_idc = np.random.choice(np.setdiff1d(mask_idc, sample_loss_index), random_mask, replace=False)

        loss_mask_idc = np.asarray(
            [
                sample_loss_index[j] + offset
                for j in range(len(sample_loss_index))
                for offset in range(lengths[j])
            ]
        )

        mask_idc = np.asarray(
            [
                mask_idc[j] + offset
                for j in range(len(mask_idc))
                for offset in range(lengths[j])
            ]
        )
        #print(loss_mask_idc)

        if len(mask_idc) == 0:
            combine_idc = loss_mask_idc
        else:
            combine_idc = np.concatenate((loss_mask_idc, mask_idc))

        mask_idcs.append(np.unique(combine_idc[combine_idc < sz]))
    min_len = min([len(m) for m in mask_idcs])
    for i, mask_idc in enumerate(mask_idcs):
        if len(mask_idc) > min_len and require_same_masks:
            mask_idc = np.random.choice(mask_idc, min_len, replace=False)
        if mask_dropout > 0:
            num_holes = np.rint(len(mask_idc) * mask_dropout).astype(int)
            mask_idc = np.random.choice(
                mask_idc, len(mask_idc) - num_holes, replace=False
            )
        mask[i, mask_idc] = True
    return mask
```