Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

Digital Object Identifier 10.1109/ACCESS.2024.0429000

Cross-Attention Reprogramming for ASR: Bridging Discrete Speech Units and Pretrained Language Models

PEI-JUN LIAO[1,2], HUNG-YI LEE[1], (Senior Member, IEEE), and HSIN-MIN WANG[2], (Senior Member, IEEE)

[1]Graduate Institute of Electrical Engineering, National Taiwan University, Taipei 10617, Taiwan
[2]Institute of Information Science, Academia Sinica, Taipei 11529, Taiwan

Corresponding author: HSIN-MIN WANG (e-mail: whm@iis.sinica.edu.tw).

ABSTRACT In automatic speech recognition (ASR), an emerging trend involves converting continuous speech features into sequences of discrete speech units (DSUs) via quantization. A key advantage of DSU representations is their compatibility with pretrained language models (PLMs), where DSUs are directly mapped to PLM token indices and the embedding layer is fine-tuned. However, this conventional strategy often relies heavily on large-scale training data to mitigate the inherent modality mismatch. In light of this, we explore a more effective way to exploit the PLM embedding dictionary. Drawing inspiration from Time-LLM, a recent time-series forecasting model, we propose a cross-attention reprogramming mechanism that incorporates codebook information from the DSU quantizer to better align the DSUs with the PLM embeddings. Compared to direct fine-tuning of PLM embeddings, our method consistently achieves improvements on the Discrete Audio and Speech Benchmark (DASB), reaching state-of-the-art performance across most DASB-style settings. We also evaluate our method on LibriSpeech-960, LibriLight-10, and Swedish, Czech, and Hungarian data from Common Voice, and observe similar trends. Notably, the proposed reprogramming method demonstrates significant gains over the fine-tuning baseline, particularly in cross-lingual and low-resource scenarios. This study proposes a new approach to using PLM embedding dictionaries in DSU-based ASR, and lays a foundation for combining speech representations with large language models in other discriminative tasks of speech processing such as speech emotion recognition and spoken question answering. The inference recipe is available at https://github.com/d06921018/CATT-RPG_for_ASR.

INDEX TERMS ASR, attention-based model, discrete speech units, embedding reprogramming, low-resource learning, pretrained language models, speech-language modeling, speech quantization

I. INTRODUCTION

Recent advances in the field of textless natural language processing (Textless NLP) [1] have shifted the focus to modeling speech by converting it into discrete symbolic representations similar to textual input. A prevailing approach in this line of work is to extract discrete speech units (DSUs) by applying vector quantization to the feature representations of self-supervised learning (SSL) speech models [2]. These DSUs are sequences of quantized indices that can be used as input to downstream language models. Automatic speech recognition (ASR), a fundamental task in speech processing, has attracted growing interest in DSU-based approaches [3]. For instance, the Discrete Speech Unit Challenge at Interspeech 2024 [4], [5] has underscored the importance of this research avenue.

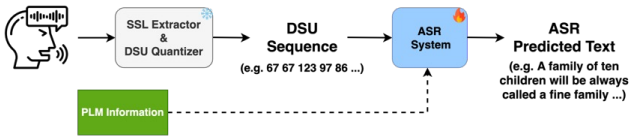


FIGURE 1: Overview of the DSU-based ASR framework. Speech is first converted into discrete speech units, which are then fed into the ASR system to generate text predictions (with PLM information).

As illustrated in Fig. 1, the current trend is to combine DSUs with powerful transformer-based architectures or integrate them with pretrained language models (PLMs), as in spoken language understanding tasks such as DUAL [6]. These studies typically adopt embedding fine-tuning, which maps DSUs to the PLM embedding space by updating the PLM's embedding

matrix, as done in DUAL and Meta AI systems [7], [8]. However, this strategy enforces a static, one-to-one mapping that struggles to bridge the substantial distributional gap between DSUs and text. Consequently, this limitation often leads to optimization instability and distortion of the PLM's pretrained embedding geometry. Since PLMs rely strongly on the structure of their embedding space to encode linguistic regularities, such distortion may degrade downstream performance and make optimization fragile, especially under limited supervision. This critical limitation necessitates a more robust, dynamic alignment mechanism. Inspired by recent developments in representation reprogramming for pretrained language models, such as Time-LLM [9], [10], we adapt a cross-attention-based [11] reprogramming technique to the DSU-based ASR setting. This line of work demonstrates that reprogramming mechanisms can effectively align non-text inputs with powerful pretrained language models [12].

From the perspective of representation learning, such a reprogramming mechanism provides a principled way to align representations between speech and language modalities. Because the distribution of DSUs differs substantially from that of textual embeddings seen during PLM pretraining, relying solely on embedding fine-tuning forces the model to learn both modality alignment and task adaptation simultaneously under limited supervision, which often leads to unstable training. In contrast, cross-attention-based reprogramming offers two key advantages: (1) Dynamic alignment – each DSU dynamically attends to the regions of the PLM embedding space that best approximate its intended textual representation; (2) Distribution bridging – cross-attention enables a richer mapping function that more naturally links DSU and PLM embedding spaces while preserving the integrity of the pretrained PLM.

Compared with embedding fine-tuning, which imposes a fixed one-to-one mapping from DSUs to PLM embeddings, and adapter-based methods that rely on shallow linear projections, our approach introduces a dedicated cross-attention reprogramming module. This module enables dynamic, input-dependent alignment by allowing each DSU to attend to the most compatible regions of the PLM embedding space, thereby alleviating the DSU-to-PLM distribution mismatch while preserving the pretrained PLM.

In this work, we demonstrate that our cross-attention reprogramming approach consistently outperforms direct fine-tuning of PLM embedding layers across a variety of ASR scenarios. Our experiments are primarily conducted on the Discrete Audio and Speech Benchmark (DASB) [13], a benchmark similar to SUPERB [14] but specifically designed to evaluate DSU-based methods. We also extend our analysis to different training resource conditions, including LibriSpeech-960 [15] and low-resource LibriLight-10 [16] settings. In addition, to evaluate the robustness of our method under cross-lingual and low-resource conditions, we perform experiments using Swedish, Czech, and Hungarian data from Common Voice [17]. Even in the case of language mismatch and insufficient supervision, our proposed method shows promising gains, validating the generalizability of cross-attention reprogramming beyond English.

Our main contributions are as follows:

- We show that cross-attention-based reprogramming consistently outperforms direct fine-tuning of PLM embedding layers on the LibriSpeech English dataset under a variety of conditions.
- We show that the proposed mechanism also achieves notable gains in cross-lingual and low-resource scenarios, indicating its potential as a general framework for speech-to-text applications beyond English.

The rest of this paper is organized as follows: Section II reviews related work. Section III introduces our proposed method, including the overall architecture and module details. Section IV describes the datasets and experimental setup. Section V presents the main results on DASB, LibriSpeech-960, LibriLight-10, and Swedish, Czech, and Hungarian data from Common Voice. Section VI provides ablation studies and analysis of extended design variants. Section VII concludes the paper and discusses possible future directions.

II. RELATED WORK

DSUs were initially used as input to PLMs in VQ-Wav2Vec [2], which has led to a growing body of research exploring this direction. This line of work differs from methods such as GSLM [18] and AudioLM [19] that focus on generative speech tasks and from models such as HuBERT [20], BEATs [21], and Wav2Seq [22] that treat DSUs as target labels for training SSL speech representations. In contrast, some studies including [23]–[33] use DSUs as input representations for tasks such as ASR, spoken language understanding, and other speech applications. While existing DSU-based ASR systems primarily rely on static direct mappings from DSUs to PLM embeddings, recent research has not explored mechanisms that enable input-dependent or dynamically evolving mappings. Inspired by the design of Time-LLM [9], we propose the first cross-attention-based reprogramming framework for DSUs, which enables mappings that change dynamically with the input, rather than relying on fixed embedding lookup. This opens up a new direction for DSU-based ASR and fills an important gap in previous static methods.

The Discrete Speech Unit Challenge at Interspeech 2024 [8], [33] formalizes this research direction by introducing evaluation metrics including character error rate (CER) and bitrate (reflecting the transmission efficiency of DSUs). In this study, we follow the DASB and use the word error rate (WER) as the main evaluation metric.

Recent research (e.g., [4]) has highlighted that DSUs can provide compression benefits in communications by encoding audio as sequences of bounded integers rather than real-valued vectors, which allows for more efficient transmission than continuous representations. However, our work does not focus on bitrate comparisons, as both the baseline and our proposed method operate on the same DSU input. Although our method involves an additional codebook for reprogramming, this component can be transmitted once in advance and does not scale with runtime data size.

Under the same DSU-based framework, [34]–[36] explored preprocessing strategies for DSU generation, including subspace projection and statistical decomposition methods grounded in linear algebra, such as principal component analysis (PCA) [37] and independent component analysis (ICA) [38], while our work focuses on how to use DSUs in downstream tasks. Methods such as DUAL and SQA-5 [39] extend DSUs to semantic understanding tasks such as spoken question answering. In contrast, our focus is on ASR.

Furthermore, as pointed out in the survey [3], most existing DSU-based ASR systems adopt a sequence-to-sequence architecture, typically employing a transformer-based decoder [11] to generate output token sequences. In this study, we follow the DASB setting and use connectionist temporal classification (CTC) [40] as the training loss and for decoding. This design can speed up training and more directly evaluate the effectiveness of our method. In addition, unlike [3], which implemented byte-pair encoded (BPE) DSU tokens, our approach directly uses raw DSUs as input. Since our model integrates representations from multiple SSL layers, we follow the DASB setup to maintain length alignment across layers without introducing additional tokenization. A recent study proposed a Particle Swarm Optimization (PSO)–based feature selection method for speech emotion recognition [41]. Although this task differs from ASR, it similarly highlights the importance of learning task-relevant representations. However, their method operates on statistically hand-crafted acoustic features extracted directly from the input speech, such as Mel-Frequency Cepstral Coefficients (MFCCs). This fundamentally differs from our dynamic representation adaptation mechanism, which operates on DSUs extracted through an SSL speech model. Our work focuses on ASR within the DASB framework, specifically improving the use of DSUs in downstream modeling. The proposed mechanism complements the benchmark by demonstrating a more effective way to transform and align DSU representations for PLM-based ASR systems. The reprogramming mechanism is general and may be extended to other speech understanding tasks, such as speech emotion recognition or spoken question answering, in the future.

III. METHODOLOGY

We extend the ASR task of DASB by introducing several architectural modifications, as shown in Fig. 2. The input waveform is first processed by a pre-trained SSL speech encoder (e.g., HuBERT-Large [20] or WavLM-Large [42]). The selected intermediate layers are passed to a pre-trained DSU quantizer to extract DSUs for downstream processing. The DSUs are transformed into continuous embeddings via an embedding module and then encoded using a pre-trained PLM, replacing the original Bi-LSTM [43] in the DASB framework [13]. The model is trained with CTC loss, and inference is performed using a CTC-based search strategy. The details of each module are presented below.

A. FEATURE EXTRACTOR AND QUANTIZER

As illustrated in Fig. 3, we follow the medium bitrate configuration of the original DASB framework and extract hidden features from six selected layers $\mathcal{L} = \{1, 3, 7, 12, 18, 23\}$ of a frozen SSL speech model (e.g., HuBERT-Large or WavLM-Large). The model directly processes the raw waveform $\mathbf{x} \in \mathbb{R}^S$, where S denotes the number of audio samples. For each target layer $l \in \mathcal{L}$, we obtain the corresponding hidden representation:

$$\mathbf{H}^{(l)} = \text{SSLModel}^{(l)}(\mathbf{x}), \quad \mathbf{H}^{(l)} \in \mathbb{R}^{T \times d}, \tag{1}$$

where T is the number of frames, and d denotes the dimensionality of the SSL features.

To obtain the corresponding DSUs, we apply six K-means [44] quantizers $\{Q^{(l)}\}_{l \in \mathcal{L}}$. Each quantizer is trained independently using an unsupervised approach on the LibriSpeech-960 dataset using features from the corresponding layer:

$$\mathbf{z}^{(l)} = Q^{(l)}(\mathbf{H}^{(l)}), \quad \mathbf{z}^{(l)} \in \{0, 1, \ldots, 999\}^T. \tag{2}$$

To distinguish DSUs from different layers, we remap each DSU index to a global vocabulary. Each layer l is assigned 1,000 indices, starting from $1000 \cdot \text{rank}(l)$:

$$\tilde{\mathbf{z}}^{(l)} = \text{Remap}\left(\mathbf{z}^{(l)} + 1000 \cdot \text{rank}(l)\right), \quad \tilde{\mathbf{z}}^{(l)} \in \mathbb{N}^T. \tag{3}$$

This remapping ensures that DSUs are assigned to unique and non-conflicting locations in the embedding space. Conflicts with reserved tokens (e.g., <s>, </s>, and <pad>) are resolved by skipping occupied indices. As a result, the last DSU index may exceed 5999.

In summary, given an input waveform $\mathbf{x} \in \mathbb{R}^S$, we extract layer-wise hidden states $\mathbf{H}^{(l)} \in \mathbb{R}^{T \times d}$, quantize them via $Q^{(l)}$ to obtain discrete units $\mathbf{z}^{(l)} \in \{0, \ldots, 999\}^T$, and apply a remapping function to produce $\tilde{\mathbf{z}}^{(l)} \in \mathbb{N}^T$, which skips reserved token indices.

B. DSU EMBEDDING MODULE

The DSU embedding module is the core of our proposed improvement scheme. In this subsection, we first introduce the baseline method of directly fine-tuning the PLM embedding layer, which employs a direct DSU-to-PLM embedding mapping. Then, we propose our contribution, the cross-attention reprogramming method, which leverages PLM's semantic space to dynamically reconstruct DSU representations, thereby enhancing alignment.

1) Directly Fine-tuning the PLM Embedding Layer

As illustrated in Fig. 4(a), we adopt a DUAL-style baseline, referred to as FT-EMB (directly fine-tuning the PLM embedding layer), where the DSUs from each selected SSL layer $l \in \mathcal{L}$ are first mapped to PLM embeddings using a shared embedding matrix $\mathbf{E}_{\mathbf{PLM}} \in \mathbb{R}^{V \times d'}$:

$$\mathbf{E}^{(l)} = \mathbf{E}_{\mathbf{PLM}}[\tilde{\mathbf{z}}^{(l)}], \quad \mathbf{E}^{(l)} \in \mathbb{R}^{T \times d'}, \tag{4}$$

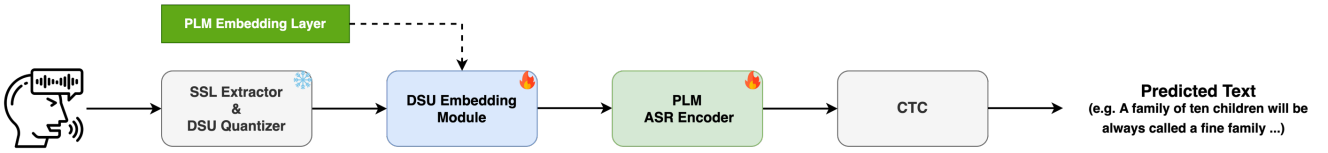


FIGURE 2: Overview of the proposed ASR framework. The input speech is first processed by the SSL Extractor and DSU Quantizer to obtain DSUs, which are then converted into embeddings through the DSU Embedding Module. Finally, the features are passed to the PLM ASR Encoder, and a CTC-based decoding mechanism is used to produce outputs for calculating training loss (training phase) or recognition accuracy (inference phase).

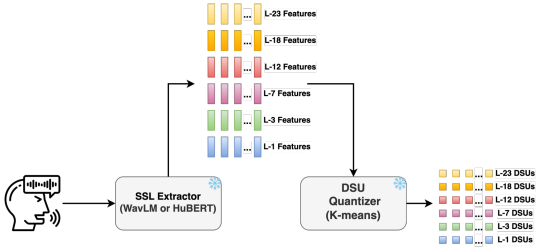


FIGURE 3: SSL feature extraction and quantization process. The waveform input is fed into a pretrained WavLM-Large or HuBERT-Large model, and features are extracted from six layers (L1, L3, L7, L12, L18, L23), each shown in a distinct color. For each layer, a 1000-cluster K-means quantizer trained on LS960 is used to produce DSUs.

where d' is the embedding dimension of the PLM ASR encoder. We apply a trainable attention mechanism, called attention multi-layer perceptron (attention MLP), to perform layer-wise fusion of embeddings at each time step. For each layer $l \in \mathcal{L}$, let $\mathbf{E}^{(l)} \in \mathbb{R}^{T\times d'}$ denote the sequence of projected embeddings. At each frame $t \in \{1,\dots,T\}$, we compute a scalar attention score by projecting the embedding vector $\mathbf{e}_t^{(l)}$ with a shared trainable vector $\mathbf{w}_{\text{att}} \in \mathbb{R}^{d'}$:

$$s_t^{(l)} = \mathbf{e}_t^{(l)} \cdot \mathbf{w}_{\text{att}}. \tag{5}$$

These scores are normalized across the six layers using a softmax function to obtain attention weights:

$$\alpha_t^{(l)} = \frac{\exp(s_t^{(l)})}{\sum_{l'\in\mathcal{L}} \exp(s_t^{(l')})}. \tag{6}$$

We then compute a fused embedding for each time step as a weighted sum of the embeddings from the six layers:

$$\mathbf{e}_{\text{fused},t} = \sum_{l\in\mathcal{L}} \alpha_t^{(l)} \cdot \mathbf{e}_t^{(l)}, \quad \mathbf{e}_{\text{fused},t} \in \mathbb{R}^{d'}. \tag{7}$$

This produces the fused embedding sequence $\mathbf{E}_{\text{fused}} = [\mathbf{e}_{\text{fused},1}; \dots; \mathbf{e}_{\text{fused},T}] \in \mathbb{R}^{T\times d'}$. The attention weights $\alpha_t^{(l)}$ are jointly learned with the downstream objective, allowing the model to dynamically attend to different SSL layers based on contextual relevance at each frame.

2) Reprogramming with PLM Embeddings

To enhance the alignment between DSU representations and PLM, we propose a cross-attention reprogramming (CATT-RPG) mechanism, as illustrated in Fig. 4(b). This method replaces the baseline's direct DSU-to-embedding mapping with a two-stage feature reprogramming process, which makes more effective use of the PLM's embedding space.

Given the DSU sequences $\tilde{\mathbf{z}}^{(l)} \in \mathbb{N}^T$ from each selected SSL layer $l \in \mathcal{L}$, we first retrieve their corresponding codeword representations using the frozen codebook $\mathbf{C_B}^{(l)} \in \mathbb{R}^{1000\times d}$ of quantizer $Q^{(l)}$, where each codeword corresponds to a centroid obtained by applying K-means clustering to the SSL layer's feature representations during quantizer training:

$$\mathbf{C}^{(l)} = \mathbf{C_B}^{(l)}[\tilde{\mathbf{z}}^{(l)}], \quad \mathbf{C}^{(l)} \in \mathbb{R}^{T\times d}. \tag{8}$$

These codeword vectors are then passed through a shared trainable linear encoding layer $\mathbf{W}_{\text{enc}} \in \mathbb{R}^{d\times d'}$ to project them into the same feature space expected by the PLM ASR encoder:

$$\hat{\mathbf{C}}^{(l)} = \mathbf{C}^{(l)} \cdot \mathbf{W}_{\text{enc}}, \quad \hat{\mathbf{C}}^{(l)} \in \mathbb{R}^{T\times d'}. \tag{9}$$

We also employ the attention MLP pooling mechanism as FT-EMB to integrate the six sets of encoded codeword features across layers. The resulting fused codeword vector sequence $\hat{\mathbf{C}}_{\text{fused}} \in \mathbb{R}^{T\times d'}$ is fed into the cross-attention reprogramming module.

To reprogram the fused codeword vectors using the semantic space of the PLM, we apply a cross-attention mechanism to the entire sequence. As illustrated in Fig. 5, we first select the first 6000 tokens (excluding the reserved special tokens) from the PLM vocabulary embedding matrix $\mathbf{E}_{\text{PLM}} \in \mathbb{R}^{|\mathcal{V}|\times d'}$ to construct a set of PLM prototype embeddings, where $\mathcal{V}$ denotes the set of all tokens in the PLM vocabulary, and $|\mathcal{V}|$ is the total vocabulary size. The selected reference subset is denoted as:

$$\mathbf{E}_{\text{PLM}}^{\text{ref}} \in \mathbb{R}^{6000\times d'}. \tag{10}$$

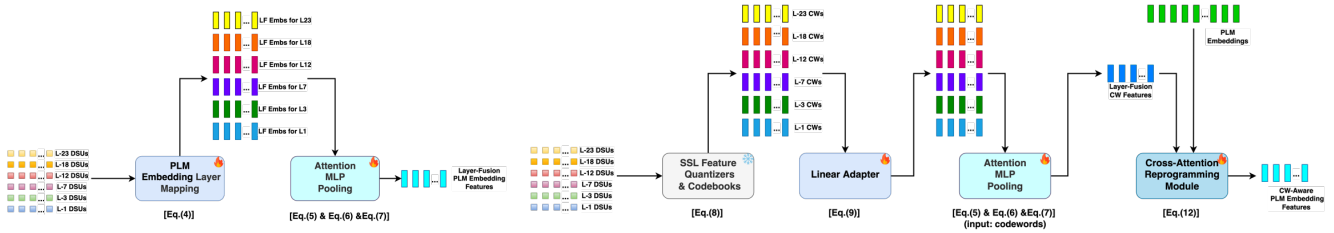
These embeddings are projected into a smaller set of trainable prototype vectors using a linear transformation:

$$\mathbf{P} = \mathbf{W}_{\text{proj}} \cdot \mathbf{E}_{\text{PLM}}^{\text{ref}} \in \mathbb{R}^{1000\times d'}, \tag{11}$$

where $\mathbf{W}_{\text{proj}} \in \mathbb{R}^{1000\times 6000}$. The fused codeword vector sequence $\hat{\mathbf{C}}_{\text{fused}} \in \mathbb{R}^{T\times d'}$ is then used as the query, while the prototype matrix $\mathbf{P}$ serves as both the key and the value in the cross-attention module as:

$$\begin{aligned} &\mathbf{Q} = \hat{\mathbf{C}}_{\text{fused}}\mathbf{W}_Q, \quad \mathbf{K} = \mathbf{P}\mathbf{W}_K, \quad \mathbf{V} = \mathbf{P}\mathbf{W}_V, \\ &\mathbf{R} = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d'}}\right)\mathbf{V}, \end{aligned} \tag{12}$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d'\times d'}$ are learnable projections. The resulting reprogrammed vector sequence $\mathbf{R} \in \mathbb{R}^{T\times d'}$ preserves the original temporal resolution and is used as input to the PLM ASR encoder.



(a) Baseline DSU embedding module (FT-EMB) (b) Proposed reprogramming-based DSU embedding module (CATT-RPG)

FIGURE 4: (a) Baseline DSU embedding module (FT-EMB): DSUs from the six layers of the SSL speech model are reassigned with indices, embedded by the PLM's embedding layer, and pooled by the attention MLP. (b) Proposed reprogramming-based DSU embedding module (CATT- RPG): DSUs are mapped to codewords, linearly projected, pooled by the attention MLP, and reprogrammed via cross-attention to form the final DSU features. [Eq. (x)] indicates the equation corresponding to the module component. In CATT-RPG, the attention MLP takes codeword features from six SSL layers as input, which differs from FT-EMB, where the corresponding PLM embedding features are used instead.

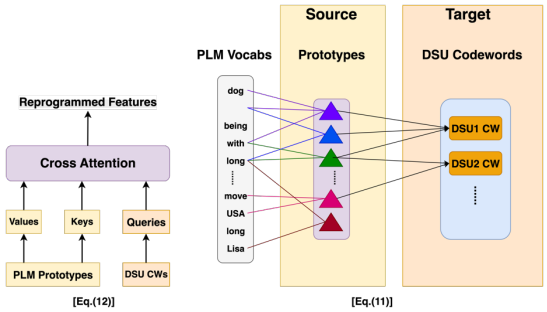


FIGURE 5: The cross-attention mechanism in reprogramming (left), where DSU codewords are used as queries and prototypes are used as keys and values. The learnable linear combination of PLM vocabulary embeddings to produce a small set of prototypes (right). [Eq. (x)] indicates the equation corresponding to the module component.

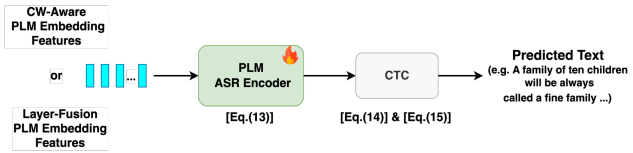


FIGURE 6: PLM ASR encoder and output configuration. The encoded DSU features are passed into the PLM ASR encoder, which is trained with CTC loss. During inference, ASR predictions are generated using CTC decoding with search, following the DASB setting. [Eq. (x)] indicates the equation corresponding to the module component.

C. PLM ASR ENCODER AND OUTPUT

The output $\mathbf{R} \in \mathbb{R}^{T \times d'}$ of the DSU embedding module is fed into the PLM ASR encoder, as shown in Fig. 6.

Following the DASB framework, we do not use preprocessing techniques such as BPE merging, which would otherwise shorten the input sequence length. As a result, the PLM ASR encoder must efficiently handle relatively long DSU sequences. To this end, we adopt Longformer-base [45] as the PLM ASR encoder architecture due to its efficient sparse attention mechanism for long-sequence modeling.

Let $\mathrm{PLM}^{(l)}(\cdot)$ denote the l-th encoder layer. We use the output from the fourth layer as our final representation:

$$\mathbf{H}_{\text{final}} = \mathrm{PLM}^{(4)}(\mathbf{R}) \in \mathbb{R}^{T \times d'}. \tag{13}$$

A linear classifier maps each time step to a distribution over the vocabulary $\mathcal{V}$:

$$\mathbf{O}_t = \mathrm{Softmax}(\mathbf{H}_{\text{final},t} \cdot \mathbf{W}_{\text{ctc}}), \quad t = 1, \ldots, T, \tag{14}$$

where $\mathbf{W}_{\text{ctc}} \in \mathbb{R}^{d' \times |\mathcal{V}|}$ is trainable. The model is trained with CTC loss:

$$\mathcal{L}_{\text{CTC}} = -\log \sum_{\pi \in B^{-1}(\mathbf{y})} \prod_{t=1}^{T} P(\pi_t \mid \mathbf{O}_t), \tag{15}$$

where π is a frame-level alignment path over the vocabulary, $B(\pi)$ denotes the standard CTC many-to-one mapping function that removes blank symbols and repeated labels to produce the target sequence $\mathbf{y}$, and $\mathbf{O}_t$ is the model output at time t [40].

The complete workflow of the proposed framework, covering both training and inference, is presented in Algorithm 1, in which the if–else statements indicate the respective branches for CATT-RPG and FT-EMB. During validation and testing, we follow the DASB setup, using CTC greedy search and CTC beam search [46], respectively. Evaluation is performed using WER and CER against reference transcripts.

IV. DATASETS AND EXPERIMENTAL SETUP

A. DATASETS

Our experiments primarily focus on the DASB ASR task, which includes both English and cross-lingual settings. For English, we follow the original DASB setting using LibriSpeech-100-Clean (LS-100) (100.6 hours) for training. To maintain distribution consistency, Dev-Clean (LS-Dev-Clean) (5.4 hours) is used as the validation set, while both Test-Clean (LS-Test-Clean) (5.4 hours) and Test-Other (LS-Test-Other) (5.1 hours) are used for testing. For the cross-lingual setting, we use Welsh and Basque from Common Voice 17.0. After filtering out samples with a transcription length of less than 3 tokens or with loading errors, the train/validation/test durations are 11.44/8.21/8.23 hours for Welsh and 116.27/26.48/22.97 hours for Basque. To analyze the impact of data size, we additionally consider LibriLight-10 (LL-10) (1 hour of clean data + 9 hours of other data) and LibriSpeech-960 (LS-960) (960.9 hours of mixed data). For both, we build a 10.7-hour validation set using a combination of Dev-Clean and

Algorithm 1 Training and Inference Procedure of DSU-based PLM-ASR (CATT-RPG and FT-EMB)

Require: SSL model, selected layers $\mathcal{L}$, quantizers $\{Q^{(l)}\}$, PLM embedding $\boldsymbol{E}_{\text{PLM}}$, reserved tokens $\mathcal{S}_{\text{resv}}$, PLM encoder, trainable params Θ
Ensure: Trained parameters Θ or predicted transcript $\hat{\boldsymbol{y}}$
for each minibatch $(\boldsymbol{x}, \boldsymbol{y})$ **do**
  Feature extraction: $\boldsymbol{H}^{(l)} \leftarrow \text{SSLModel}^{(l)}(\boldsymbol{x})$
  Quantization: $\boldsymbol{z}^{(l)} \leftarrow Q^{(l)}(\boldsymbol{H}^{(l)})$
  Remap: $\tilde{\boldsymbol{z}}^{(l)} \leftarrow \text{Remap}\big(\boldsymbol{z}^{(l)} + 1000 \cdot \text{rank}(l)\big)$
  if CATT-RPG branch **then**
    Retrieve codewords $\boldsymbol{C}^{(l)} \leftarrow \boldsymbol{C}_B^{(l)}[\tilde{\boldsymbol{z}}^{(l)}]$
    Encode and Fuse (via Attn-MLP) $\hat{\boldsymbol{C}}^{(l)} \leftarrow \boldsymbol{C}^{(l)}\boldsymbol{W}_{\text{enc}}$, $\hat{\boldsymbol{C}}_{\text{fused}} \leftarrow \text{Fuse}(\{\hat{\boldsymbol{C}}^{(l)}\}_l)$
    Project to PLM space $\boldsymbol{P} = \boldsymbol{W}_{\text{proj}}\boldsymbol{E}_{\text{PLM}}^{\text{ref}}$
    Cross-attend $\boldsymbol{R} = \text{Attention}(\hat{\boldsymbol{C}}_{\text{fused}}, \boldsymbol{P}, \boldsymbol{P})$
  else
    FT-EMB branch:
    Retrieve PLM embeddings $\boldsymbol{E}^{(l)} = \boldsymbol{E}_{\text{PLM}}[\tilde{\boldsymbol{z}}^{(l)}]$,
    Fuse (via Attn-MLP) $\boldsymbol{E}_{\text{fused}} \leftarrow \text{Fuse}(\{\boldsymbol{E}^{(l)}\}_l) \Rightarrow \boldsymbol{R} = \boldsymbol{E}_{\text{fused}}$
  end if
  Obtain contextual output $\boldsymbol{H}_{\text{final}} = \text{PLM}^{(4)}(\boldsymbol{R})$
  if mode = training **then**
    Compute posterior $\boldsymbol{O}_t = \text{Softmax}(\boldsymbol{H}_{\text{final},t}\boldsymbol{W}_{\text{ctc}})$
    Evaluate CTC loss $\mathcal{L}_{\text{CTC}} = \text{CTC}(\boldsymbol{O}_{1:T}, \boldsymbol{y})$
    Update parameters Θ via backpropagation
  else if mode = validation **then**
    Decode hypothesis $\hat{\boldsymbol{y}} = \text{CTC-Greedy}(\boldsymbol{H}_{\text{final}})$
  else
    Decode hypothesis $\hat{\boldsymbol{y}} = \text{CTC-BeamSearch}(\boldsymbol{H}_{\text{final}})$
  end if
end for

Dev-Other, following the SSL-Selection-Fusion [47] and R-EKD [48] protocols. The test sets are consistent with the DASB task described above. To further explore combinations of low-resource and cross-lingual settings (such as the case of Welsh), we include Swedish, Hungarian, and Czech data from Common Voice. Specifically, Swedish data is taken from Common Voice 21.0 with train/validation/test durations of 8.91/5.91/6.73 hours, Hungarian data from Common Voice 21.0 with 54.69/16.88/17.76 hours, and Czech data from Common Voice 22.0 with 28.63/11.83/11.80 hours after filtering.

B. EXPERIMENTAL SETUP

The baseline method FT-EMB directly fine-tunes the PLM's embedding layer, while our proposed method CATT-RPG applies cross-attention reprogramming to derive DSU features. In addition to FT-EMB, we also include another baseline method (called Adapter) in our experiments. This method inserts a small linear adapter ($\mathbf{W} \in \mathbb{R}^{d' \times d'}$, $d' = 768$) between the PLM's embedding layer and the PLM's first transformer encoder layer. During downstream ASR fine-tuning, the adapter and the PLM's transformer encoder are updated, but the PLM's embedding layer is frozen. We use a batch size of 4. For DASB, LL-10, Hungarian, Czech, and Swedish tasks, we train for 45 epochs; due to the extensive dataset size, we train for 25 epochs on the LS-960 task.

We follow the DASB scheduler settings. The learning rate is set to 0.0001 to accommodate the Longformer PLM (embedding size: 768). We re-implemented the DASB Bi-LSTM model with the same hyperparameters in DASB [13] (learning rate: 0.0002, embedding size: 1024). Our replication results are slightly different from those in [13], which may be due to the difference in the experimental environment. Since WavLM and HuBERT yield the best performance in DSU-based ASR in the DASB setting [13], we use them as feature extractors to generate DSUs. The hidden feature vectors extracted through HuBERT-Large and WavLM-Large, as well as the associated codeword vectors retrieved from the codebook of each quantizer, all have a dimension of 1024. To reduce computational cost, we pre-extract and store DSUs before downstream training. Our implementation environment includes PyTorch 2.5.1+cu124, HuggingFace Transformers [49] 4.47.1, SpeechBrain [50] 1.0.2, and an NVIDIA GeForce RTX 3090 GPU for model training.

V. EXPERIMENTS

Before presenting the main results and comparative analyses, we first conduct preliminary experiments in Sections V-A and V-B to provide an evidence-based justification for our core design choices and hyperparameters. These preliminary analyses determine the optimal output layer for the PLM ASR encoder and the appropriate number of prototypes required for the CATT-RPG method.

A. ANALYSIS OF THE PLM ASR ENCODER

As described in the ''PLM ASR Encoder and Output'' subsection, we only use the first four layers of the pretrained Longformer as the PLM ASR encoder and take its fourth layer as output. This choice is based on experiments where the FT-EMB model is trained on LS-100 and evaluated on LS-Dev-Clean. We compare the performance of Longformer layers 2, 4, 6, 8, 10, and 12, using WER as the selection criterion. As illustrated in Fig. 7, the fourth layer achieves the lowest WER, while the sixth layer also yields competitive performance, regardless of using HuBERT or WavLM as the SSL feature extractor. Considering the trade-off between performance and parameter count, we use the first four layers of the pretrained Longformer as the PLM ASR encoder and take the output of the fourth layer, which provides a balanced contextual representation while maintaining computational efficiency for downstream ASR modeling.

B. ANALYSIS OF THE NUMBER OF PROTOTYPES IN REPROGRAMMING

In our CATT-RPG method, the number of prototypes is initially set to 1000 following the configuration used in Time-LLM [9]. We further evaluate the impact of the number of prototypes

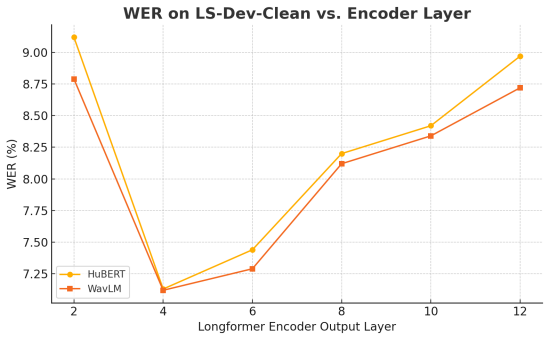


FIGURE 7: WER of FT-EMB using different numbers of pre-trained Longformer layers as PLM ASR encoder on LS-100. In each case, the last layer is used as the output. Clearly, the fourth layer achieves the best performance.

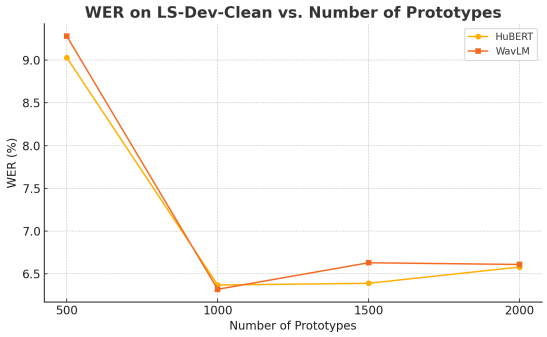


FIGURE 8: WER of CATT-RPG under different prototype sizes (500–2000) on LS-100. Both HuBERT and WavLM perform best at 1000, supporting its selection as the final setting.

on performance by varying it between 500, 1000, 1500, and 2000, while keeping all other conditions fixed. The models are trained on LS-100 and evaluated on LS-Dev-Clean. As shown in Fig. 8, HuBERT achieves the lowest WER at 1000, followed by 1500, while WavLM performs best at 1000 by a significant margin. In contrast, when only 500 prototypes are used, both models experience significant performance degradation. These results indicate that the number of prototypes has a significant impact on the effectiveness of CATT-RPG. Considering the trade-off between performance and the number of parameters, we select 1000 as the prototype size. Notably, this value lies near the intrinsic upper bound of the DSU feature space (1024 dimensions), which represents the maximum number of non-zero eigen components in the codeword matrix ($\approx 6000 \times 1024$). This alignment suggests that the chosen prototype size adequately captures the representational capacity of the underlying feature manifold while avoiding unnecessary redundancy.

C. RESULTS ON THE DASB ASR TASKS

We compare the performance of our proposed CATT-RPG with the main baseline FT-EMB on the DASB ASR task. In addition to using the results provided in the DASB paper [13], we also reproduce the original implementation of the Bi-LSTM encoder according to the released recipe. To facilitate reproducibility, FT-EMB is configured to use the first 6000 indices in order, corresponding to 1000 DSUs from each of the six SSL layers. CATT-RPG follows the same DSU configuration to ensure a fair comparison and consistent alignment with the PLM embeddings. We also implement a variant of CATT-RPG that utilizes the full PLM vocabulary (excluding special tokens) to enable more comprehensive comparison and analysis.

As shown in Table 1, our CATT-RPG method consistently outperforms the FT-EMB baseline in the LS-Test-Clean, LS-Test-Other, Welsh, and Basque ASR tasks. With HuBERT and WavLM as SSL feature extractors, we observe WER reductions of 9.58% (from 7.41% to 6.70%), 6.84% (from 17.41% to 16.22%), 5.77% (from 61.13% to 57.60%), and 3.10% (from 22.57% to 21.87%), and 11.42% (from 7.44% to 6.59%), 9.90% (from 17.18% to 15.48%), 9.06% (from 61.48% to 55.91%), and 3.06% (from 20.29% to 19.67%), respectively. These results demonstrate the effectiveness of our method on these DASB benchmark ASR tasks. Furthermore, the Adapter baseline, which updates only a small $d' \times d'$ layer while keeping the PLM embedding layer frozen, consistently underperforms the FT-EMB baseline on all four DASB test sets, except for the WER of LS-Test-Other.

Several other observations can be made from Table 1. Interestingly, CATT-RPG with the full PLM vocabulary demonstrates superior performance over FT-EMB across most evaluation scenarios, including a state-of-the-art (SOTA) Basque CER on WavLM. This suggests that leveraging the entire vocabulary holds promise for enhancing cross-lingual generalization, pointing to a potential direction for future research. In addition, for Welsh (with only 11.44 hours of training data), both the official and our reproduced Bi-LSTM-based models outperform CATT-RPG and FT-EMB. This suggests that incorporating a PLM is not always beneficial in low-resource and cross-lingual settings. We will explore this observation further in the next subsection. Finally, we observe that the CER results align well with the WER results. CATT-RPG consistently achieves SOTA performance in all test sets except Welsh. The best results on the validation sets shown in Table 2 also support the same conclusion.

D. RESULTS ON LIBRISPEECH-960 AND LIBRILIGHT-10

In our second set of experiments, we evaluate the impact of training data size on ASR performance on English datasets by comparing the performance of models trained on different training sets on LS-Test-Clean and LS-Test-Other. As shown in Table 3, while CATT-RPG still maintains SOTA performance, its WER reduction over FT-EMB becomes smaller in most cases. Using HuBERT, for the model trained on LS-960, the WER on LS-Test-Clean and LS-Test-Other decreases by 1.57% (from 3.81% to 3.75%) and 0.65% (from 9.29% to 9.23%), respectively; for the model trained on LL-10, the WER decreases by 0.84% (from 21.34% to 21.16%) and 1.59% (from 31.38% to 30.88%), respectively. The reduction in WER with WavLM is more significant; in particular, for the model trained on LL-10, the WER on LS-Test-Clean and LS-Test-Other decreases by 8.54% (from 21.44% to 19.61%) and 3.64% (from 31.04% to 29.91%), respectively. This suggests

TABLE 1: ASR performance on the test sets of different DASB ASR tasks. "Bi-LSTM [13]" denotes the results in [13]. "Bi-LSTM" denotes our replicated results. "TFS" denotes train from scratch, which is the embedding module setting used in DASB. "FT-EMB" is the baseline that fine-tunes the PLM embedding layer directly. "Adapter" is the baseline that uses a learnable linear adapter layer after the frozen PLM embedding layer. "CATT-RPG" is our proposed cross-attention reprogramming method. Overall, CATT-RPG achieves state-of-the-art results on DASB in all test conditions except the Welsh task. Underlined values indicate the best performance for each SSL feature extractor. **Bold** values indicate that CATT-RPG outperforms the FT-EMB baseline. Overall, CATT-RPG consistently achieves state-of-the-art results across both SSL extractors.

SSL Extractor	ASR Encoder	Unit Embedding	PLM Vocab Size	WER (%) ↓				CER (%) ↓			
				English		Cross-lingual		English		Cross-lingual	
				Clean	Other	Welsh	Basque	Clean	Other	Welsh	Basque
HuBERT	Bi-LSTM [13]	TFS	–	7.91	18.95	54.77	23.63	–	–	–	–
	Bi-LSTM			8.14	19.55	56.55	23.19	2.30	6.64	17.57	4.46
	Longformer	FT-EMB	6000	7.41	17.41	61.13	22.57	2.08	5.95	19.09	4.22
		Adapter	6000	7.57	17.95	63.60	23.71	2.13	6.16	20.08	4.51
		CATT-RPG	50260	**6.99**	**17.02**	**59.12**	24.40	**1.96**	**5.81**	**18.20**	4.60
			6000	**6.70**	**16.22**	**57.60**	**21.87**	**1.88**	**5.55**	**17.63**	**4.10**
WavLM	Bi-LSTM [13]	TFS	–	8.52	20.35	54.22	22.06	–	–	–	–
	Bi-LSTM			7.99	18.57	54.74	20.80	2.24	6.31	16.39	3.76
	Longformer	FT-EMB	6000	7.44	17.18	61.48	20.29	2.08	5.80	18.53	3.63
		Adapter	6000	7.49	16.97	62.12	21.26	2.14	5.81	19.07	3.83
		CATT-RPG	50260	**6.90**	**16.49**	63.00	**20.02**	**1.97**	**5.61**	19.87	**3.51**
			6000	**6.59**	**15.48**	**55.91**	**19.67**	**1.87**	**5.35**	**16.60**	**3.56**

TABLE 2: ASR performance on the validation sets of different DASB ASR tasks. In the 6K vocabulary setting, CATT-RPG always outperforms FT-EMB, but in the 50K vocabulary setting, CATT-RPG does not always outperform FT-EMB. Underlined: the best performance for each SSL feature extractor; **bold**: CATT-RPG better than FT-EMB.

SSL Extractor	ASR Encoder	Unit Embedding	PLM Vocab Size	WER (%) ↓			CER (%) ↓		
				English	Welsh	Basque	English	Welsh	Basque
HuBERT	Bi-LSTM	TFS	–	7.97	56.20	22.34	2.25	16.85	4.19
	Longformer	FT-EMB	6000	7.13	60.69	21.98	2.04	18.28	3.97
		Adapter	6000	7.54	63.56	23.16	2.14	19.40	4.30
		CATT-RPG	50260	**6.65**	**58.58**	**23.88**	**1.91**	**17.41**	4.38
			6000	**6.37**	**57.15**	**21.31**	**1.82**	**16.75**	**3.87**
WavLM	Bi-LSTM	TFS	–	7.61	53.77	20.28	2.20	15.64	3.60
	Longformer	FT-EMB	6000	7.12	60.72	20.06	2.06	17.90	3.49
		Adapter	6000	7.12	61.89	21.22	2.07	18.42	3.72
		CATT-RPG	50260	**6.74**	62.81	**19.56**	**1.94**	19.26	**3.34**
			6000	**6.32**	**55.96**	**19.51**	**1.84**	**15.97**	**3.45**

TABLE 3: ASR performance of models trained on LS-960 and LL-10 and tested on LS-Test-Clean and LS-Test-Other. Overall, in the individual cases of the two SSL feature extractors, CATT-RPG consistently achieves SOTA performance. Underlined: the best performance for each SSL feature extractor; **bold**: CATT-RPG better than FT-EMB.

SSL Extractor	ASR Encoder	Unit Embedding	WER (%) ↓				CER (%) ↓			
			LS-960		LL-10		LS-960		LL-10	
			Clean	Other	Clean	Other	Clean	Other	Clean	Other
HuBERT	Bi-LSTM	TFS	4.31	10.95	21.85	33.05	1.20	3.70	6.39	11.49
	Longformer	FT-EMB	3.81	9.29	21.34	31.38	1.05	3.17	6.24	10.84
		Adapter	4.19	10.07	22.09	31.76	1.15	3.40	6.55	11.07
		CATT-RPG	**3.75**	**9.23**	**21.16**	**30.88**	**1.02**	**3.12**	**6.21**	**10.49**
WavLM	Bi-LSTM	TFS	4.26	10.78	20.49	31.50	1.17	3.60	5.97	10.87
	Longformer	FT-EMB	3.82	9.24	21.44	31.04	1.04	3.14	6.28	10.55
		Adapter	4.01	9.73	21.77	31.52	1.09	3.30	6.35	10.84
		CATT-RPG	**3.76**	**9.08**	**19.61**	**29.91**	**1.00**	**3.07**	**5.86**	**10.50**

TABLE 4: ASR performance of the models trained on LS-960 and LL-10 on the validation set. Underlined: the best performance for each SSL feature extractor; **bold**: CATT-RPG better than FT-EMB.

SSL Extractor	ASR Encoder	Unit Embedding	WER (%) ↓		CER (%) ↓	
			LS-960	LL-10	LS-960	LL-10
HuBERT	Bi-LSTM	TFS	7.38	27.31	2.44	8.80
	Longformer	FT-EMB	6.26	25.76	2.10	8.28
		Adapter	6.75	26.33	2.26	8.57
		CATT-RPG	**6.16**	**25.42**	**2.06**	**8.26**
WavLM	Bi-LSTM	TFS	7.29	25.58	2.40	8.34
	Longformer	FT-EMB	6.39	25.77	2.13	8.32
		Adapter	6.78	26.28	2.25	8.49
		CATT-RPG	**6.32**	**24.17**	**2.10**	**8.02**

that WavLM-based DSUs are more effective in low-resource training settings. Under all training data sizes (LS-960 and LL-10) and feature extractors (HuBERT and WavLM), the Adapter baseline consistently performs slightly worse than the FT-EMB baseline. Interestingly, under low-resource training conditions (i.e., LL-10), the Bi-LSTM-based model performs on par with PLM-based models, and sometimes even outperforms FT-EMB. We will further analyze this observation in the next subsection. In contrast, under rich-resource training conditions (i.e., LS-960), the Bi-LSTM-based model clearly performs worse than PLM-based models. However, the performance gap

TABLE 5: ASR performance on the Welsh, Swedish, Czech, and Hungarian tasks in Common Voice. Underlined: best performance for each SSL feature extractor; **bold**: CATT-RPG performs better than FT-EMB.

SSL Extractor	ASR Encoder	Unit Embedding	WER (%) Welsh Val	Welsh Test	Swedish Val	Swedish Test	Czech Val	Czech Test	Hungarian Val	Hungarian Test
HuBERT	Bi-LSTM	TFS	56.20	56.55	68.77	68.11	50.78	52.24	36.39	37.23
	Longformer	FT-EMB	60.69	61.13	73.09	72.84	54.41	55.68	39.03	39.52
		Adapter	63.56	63.60	73.78	73.15	55.99	56.80	39.70	40.31
		CATT-RPG	**57.15**	**57.60**	**70.58**	**70.26**	**47.92**	**48.98**	**35.56**	**36.17**
WavLM	Bi-LSTM	TFS	53.77	54.74	66.84	66.18	46.98	48.49	35.01	35.31
	Longformer	FT-EMB	60.72	61.48	70.82	70.74	49.16	50.22	36.10	36.35
		Adapter	61.89	62.12	71.99	71.40	51.83	52.43	37.70	38.09
		CATT-RPG	**55.96**	**55.91**	**66.21**	**65.96**	**44.48**	**45.68**	**32.44**	**32.77**
SSL Extractor	ASR Encoder	Unit Embedding	CER (%) Welsh Val	Welsh Test	Swedish Val	Swedish Test	Czech Val	Czech Test	Hungarian Val	Hungarian Test
HuBERT	Bi-LSTM	TFS	16.85	17.57	24.04	23.72	12.58	13.42	7.75	8.34
	Longformer	FT-EMB	18.28	19.09	25.42	25.14	13.35	14.16	8.22	8.69
		Adapter	19.40	20.08	26.22	25.80	13.98	14.75	8.42	8.97
		CATT-RPG	**16.75**	**17.63**	**24.32**	**24.02**	**11.43**	**12.22**	**7.29**	**7.80**
WavLM	Bi-LSTM	TFS	15.64	16.39	22.38	22.08	11.07	11.85	7.15	7.62
	Longformer	FT-EMB	17.90	18.53	23.77	23.61	11.38	12.03	7.29	7.69
		Adapter	18.42	19.07	24.73	24.38	12.20	12.81	7.70	8.12
		CATT-RPG	**15.97**	**16.60**	**21.86**	**21.67**	**10.05**	**10.64**	**6.39**	**6.82**

between CATT-RPG and FT-EMB also narrows in this case. This indicates that FT-EMB is already able to provide strong performance when sufficient training data is available. These findings suggest that with WavLM DSUs, the main advantage of CATT-RPG is in low-resource training scenarios, where it can provide more notable improvements. Similar trends are observed for the performance on the validation set shown in Table 4.

E. CROSS-LINGUAL AND LOW-RESOURCE ANALYSIS

We further investigate low-resource training and cross-lingual conditions (the training language differs from the PLM pre-training data). We chose Swedish, Czech, and Hungarian from Common Voice for our experiments; their training sets contain 8.91, 28.63, and 54.69 hours of speech, respectively, due to their use of the Latin alphabet and the size of the processed datasets. The best results on the validation set and the corresponding results on the test set are shown in Table 5. For these tasks, CATT-RPG outperforms FT-EMB on both the validation and test sets, similar to the results observed in the Welsh task. For the Swedish task, WER decreases by 3.43% (from 73.09% to 70.58%) and 3.54% (from 72.84% to 70.26%) when using HuBERT, and by 6.51% (from 70.82% to 66.21%) and 6.76% (from 70.74% to 65.96%) when using WavLM. For Czech, WER decreases by 12.03% (from 55.68% to 48.98%) with HuBERT and by 9.04% (from 50.22% to 45.68%) with WavLM. For Hungarian, WER decreases by 8.48% (from 39.52% to 36.17%) with HuBERT and by 9.85% (from 36.35% to 32.77%) with WavLM. These consistent improvements across the three languages demonstrate the robustness of the CATT-RPG approach under low-resource and cross-lingual conditions. In particular, CATT-RPG based on WavLM achieves SOTA performance across all three languages other than Welsh. However, in the Welsh and Swedish tasks, CATT-RPG does not consistently outperform the Bi-LSTM-based model, which we attribute to the limited amount of training data (around 10 hours) available for these languages. This finding supports our previous hypothesis that, in settings with extremely limited training data, the benefit of introducing PLM is limited if the target language is different from the training language of the PLM. Furthermore, under this experimental setup, the Adapter baseline consistently performs worse than the FT-EMB baseline.

TABLE 6: Comparison of the number of parameters for FT-EMB (directly fine-tuning the PLM embedding layer), Adapter (learning an adapter layer for the frozen PLM embedding layer), and CATT-RPG (cross-attention reprogramming).

FT-EMB		**Adapter**		**CATT-RPG**	
Trainable	Total	Trainable	Total	Trainable	Total
83.6 M	83.6 M	79.0 M	83.6 M	87.0 M	93.2 M

F. COMPARISON OF TRAINABLE PARAMETERS

To provide a clear comparison of model complexity, Table 6 summarizes the parameter counts of the FT-EMB, Adapter, and CATT-RPG models, excluding the SSL encoder. The Adapter baseline has the fewest trainable parameters (79.0M), but this parameter efficiency comes at the cost of inferior performance. Compared to FT-EMB, CATT-RPG introduces a moderate increase in trainable parameters (87.0M) and incurs additional computation from the cross-attention reprogramming module, whose complexity scales linearly with the input length as $O(T \cdot P)$, where P denotes the number of prototypes. This overhead remains manageable in practice and preserves the efficiency advantages of the Longformer backbone.

Furthermore, results of all three experimental settings (DASB ASR, training data size analysis, and cross-lingual analysis) consistently demonstrate that the Adapter baseline is less effective than the FT-EMB baseline. This observation confirms that, in the DSU-based setting, FT-EMB is more effective than relying solely on a linear adapter layer. Therefore, in the subsequent detailed analysis and comparisons, we focus exclusively on the FT-EMB baseline.

TABLE 7: WER and CER on Welsh and Swedish using different DSU embedding modules. **Bold**: performance improved after fusion; underlined: the best performance for each SSL extractor; +: output-sum fusion before CTC.

SSL Extractor	DSU Embedding	WER		CER	
		Welsh	Swedish	Welsh	Swedish
HuBERT	Bi-LSTM (0)	56.55	68.11	17.57	23.72
	FT-EMB (1)	61.13	72.84	19.09	25.14
	CATT-RPG (2)	57.60	70.26	17.63	24.02
	(0) + (1)	56.67	68.40	**17.55**	23.76
	(0) + (2)	**55.12**	**67.09**	**17.06**	**23.27**
WavLM	Bi-LSTM (0)	54.74	66.18	16.39	22.08
	FT-EMB (1)	61.48	70.74	18.53	23.61
	CATT-RPG (2)	55.91	65.96	16.60	21.67
	(0) + (1)	**54.25**	**65.17**	**16.20**	**21.61**
	(0) + (2)	**53.74**	**64.75**	**16.07**	**21.40**

TABLE 8: Ablation study results of CATT-RPG trained on the LS-100 training set and tested on the LibriSpeech test sets. "AvgPool" means replacing attention MLP pooling with average pooling. "TFS-DSU" means replacing DSU-to-codeword mapping with training the DSU embedding layer from scratch. "w/o Proto" means removing the prototype linear layer in cross-attention reprogramming. "w/o RP" means removing the cross-attention reprogramming module and substituting it with a linear layer adapter. **Bold**: the best performance for each SSL feature extractor

SSL Extractor	Ablation	WER (%) ↓		CER (%) ↓	
		Clean	Other	Clean	Other
HuBERT	CATT-RPG	**6.70**	**16.22**	**1.88**	**5.55**
	AvgPool	6.72	16.48	1.89	5.64
	TFS-DSU	7.48	18.18	2.14	6.31
	w/o Proto	7.60	17.26	2.43	6.15
	w/o RP	13.61	26.52	3.52	8.38
WavLM	CATT-RPG	**6.59**	**15.48**	**1.87**	**5.35**
	AvgPool	6.71	16.38	1.90	5.52
	TFS-DSU	7.52	17.02	2.16	6.00
	w/o Proto	6.96	16.23	1.97	5.55
	w/o RP	17.46	31.14	4.44	9.75

G. FUSION-BASED COMPLEMENTARITY OF CATT-RPG AND BI-LSTM IN LOW-RESOURCE CROSS-LINGUAL ASR

To further investigate the model's behavior in extremely low-resource and cross-lingual scenarios, we use Welsh and Swedish as representative cases, fusing CATT-RPG and FT-EMB with Bi-LSTM respectively, comparing and examining their complementarity. Specifically, we compare performance before and after module fusion, where module fusion refers to summing the outputs of two modules before computing the CTC score. As shown in Table 7, CATT-RPG consistently benefits from module fusion with Bi-LSTM under all settings, whereas FT-EMB fused with Bi-LSTM even leads to performance degradation when using HuBERT as the extractor. Notably, CATT-RPG combined with Bi-LSTM achieves SOTA performance on both Welsh and Swedish, validating our hypothesis that CATT-RPG exhibits greater complementarity with Bi-LSTM (operating directly on DSU embeddings) than FT-EMB, thereby demonstrating the value of our proposed CATT-RPG.

VI. ABLATION STUDY AND DISCUSSION

A. ABLATION STUDY

Our ablation study evaluates several variants of CATT-RPG in the DASB LS-100 ASR setting by modifying its key components:

(i) AvgPool: Replace attention MLP pooling (see Fig. 4(b)) with average pooling of the codeword features of the six DSU layers.
(ii) TFS-DSU: Replace the codeword embedding and linear adapter (see Fig. 4(b)) with a linear embedding layer trained from scratch, consistent with the original DASB setting.
(iii) w/o RP: Remove the cross-attention reprogramming module (see Fig. 4(b)) and substitute it with a linear layer adapter.
(iv) w/o Proto: Remove the prototype linear layer from the reprogramming module (see Fig. 5), and directly use the first 6000 tokens (excluding reserved special tokens) from the PLM vocabulary in cross-attention reprogramming.

As shown in Table 8, as expected, substituting the cross-attention reprogramming module with a linear layer adapter (w/o RP) results in the largest performance degradation, indicating a significant mismatch between the codeword features and the PLM ASR encoder.

Removing the prototype linear layer in the cross-attention reprogramming module (w/o Proto), replacing the codeword embedding and linear adapter with a linear embedding layer trained from scratch (TFS-DSU), and replacing attention MLP pooling with average pooling (AvgPool) also lead to performance degradation. Average pooling (AvgPool) is a less detrimental alternative, since the DSUs of all six SSL layers are fully accounted for during model training.

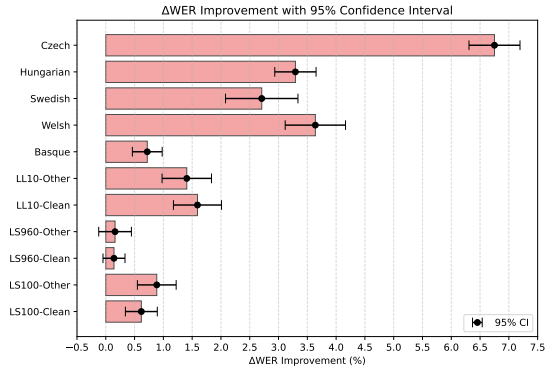
B. CONFIDENCE INTERVAL ANALYSIS FOR WER AND CER IMPROVEMENTS

In this section, we validate the stability of the improvement achieved by CATT-RPG over FT-EMB for each test utterance based on the mean and 95% confidence interval (CI) of the WER and CER improvement differences. For each test utterance, the improvement difference is calculated by subtracting the CATT-RPG WER/CER from that of the baseline FT-EMB:

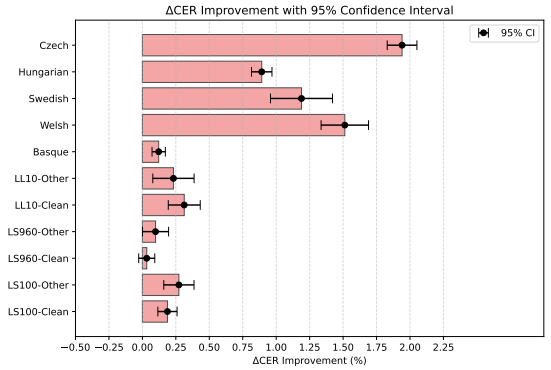
$$\Delta\text{WER} = \text{WER}_{\text{FT-EMB}} - \text{WER}_{\text{CATT-RPG}}, \quad (16)$$

$$\Delta\text{CER} = \text{CER}_{\text{FT-EMB}} - \text{CER}_{\text{CATT-RPG}}. \quad (17)$$

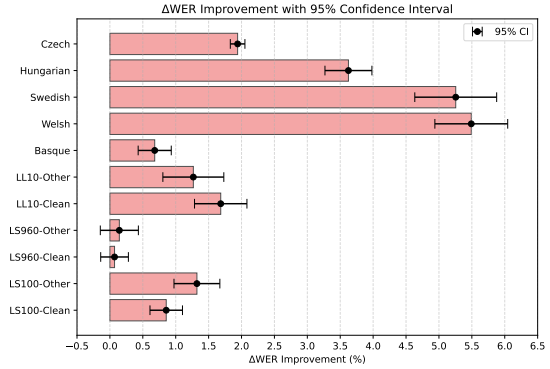
A positive value indicates that CATT-RPG outperforms FT-EMB (since lower WER/CER values are better). If the entire confidence interval lies above zero, it implies higher stability of the proposed method. Figs. 9(a) and 9(b) show the average WER and CER improvement differences when using HuBERT as the feature extractor, while Figs. 9(c) and 9(d) present the results with WavLM as the feature extractor. We observe that except for the LS-960 set, whose lower bound falls below zero, and the WavLM-based LL-10 CER case, where the lower bound is slightly below zero, all other intervals remain strictly positive. This demonstrates the robustness of our method. Moreover, the LS-960 results further confirm our initial



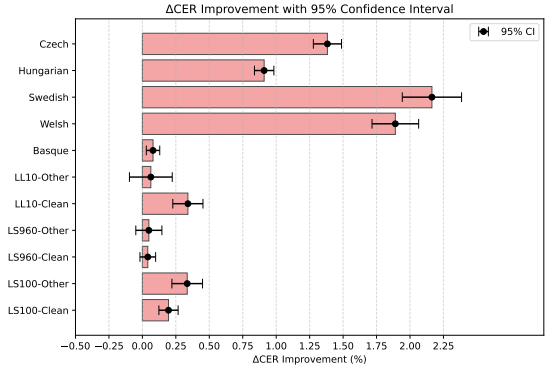
(a) HuBERT-DSU ΔWER 95% CI



(b) HuBERT-DSU ΔCER 95% CI



(c) WavLM-DSU ΔWER 95% CI



(d) WavLM-DSU ΔCER 95% CI

FIGURE 9: Statistical analysis of the improvements achieved by CATT-RPG over FT-EMB. Subfigures (a) and (b) use HuBERT as the feature extractor (denoted as HuBERT-DSU), while (c) and (d) use WavLM as the feature extractor (denoted as WavLM-DSU). For each test utterance, the ΔWER and ΔCER are calculated as the differences between FT-EMB and CATT-RPG, and their averages with 95% confidence intervals are analyzed. A more consistently positive ΔWER/ΔCER indicates stronger robustness. The horizontal axis represents the ΔWER or ΔCER values, and the vertical axis corresponds to the ASR tasks. For example, "LS100-Clean" indicates that the model is trained on the LS-100 training subset and evaluated on the corresponding LS-Test-Clean set.

hypothesis that when sufficient training data are available, the disadvantage of FT-EMB becomes less pronounced.

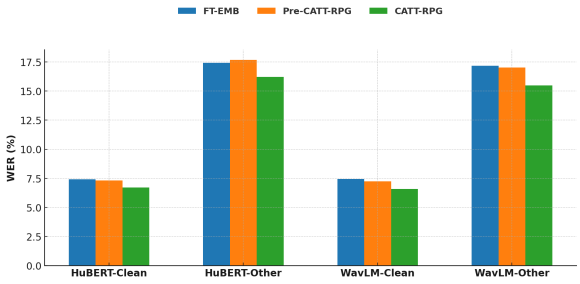


FIGURE 10: WER (%) of four "SSL-Test" scenarios using different DSU embedding methods. The models are trained on the LS-100 training set. Exact values by (FT-EMB (baseline), Pre-CATT-RPG, and CATT-RPG (proposed)) are as follows: (7.41 / 7.31 / 6.70) for HuBERT-Clean, (17.41 / 17.67 / 16.22) for HuBERT-Other, (7.44 / 7.24 / 6.59) for WavLM-Clean, and (17.18 / 17.02 / 15.48) for WavLM-Other.

C. PRE-CATT-RPG VERSUS CATT-RPG

In our initial design, the codeword features extracted from each layer of the SSL model are individually reprogrammed before pooling. We refer to this strategy as Pre-CATT-RPG. In this set of experiments, we compare it with the proposed CATT-RPG

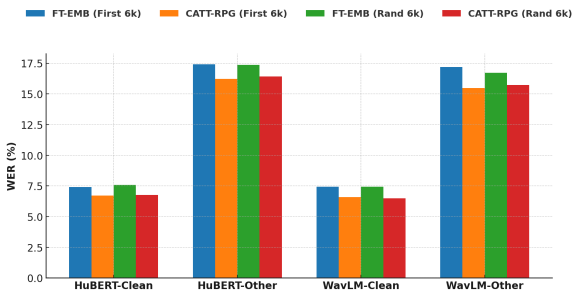


FIGURE 11: WER (%) of four "SSL-Test" scenarios under two types of vocabulary settings: first 6000 and random 6000. The models are trained on the LS-100 training set. Exact values by (FT-EMB (First 6k), CATT-RPG (First 6k), FT-EMB (Rand 6k), and CATT-RPG (Rand 6k)) are as follows: (7.41 / 6.70 / 7.58 / 6.76) for HuBERT-Clean, (17.41 / 16.22 / 17.37 / 16.41) for HuBERT-Other, (7.44 / 6.59 / 7.43 / 6.50) for WavLM-Clean, and (17.18 / 15.48 / 16.73 / 15.72) for WavLM-Other.

and the FT-EMB baseline on the DASB LS-100 task. Fig. 10 shows the WER results tested on LS-Test-Clean and LS-Test-Other using DSUs extracted from HuBERT and WavLM. It is obvious that CATT-RPG outperforms Pre-CATT-RPG in all settings. Notably, on Test-Other, when using HuBERT, Pre-CATT-RPG even performs worse than FT-EMB. These findings highlight the advantage of our proposed CATT-RPG

design, where pooling precedes the reprogramming process.

D. SELECTION OF PLM TOKENS FOR DSUS

In the above experiments, we used the first 6000 tokens in PLM after excluding special tokens to represent DSUs. Here, we compare this method with a set of random 6000 vocabulary indices (excluding special tokens) [6] on the LS-100 task. As shown in Fig. 11, selecting the first 6k performs slightly better than randomly selecting 6k, but the performance difference is small. It is worth noting that CATT-RPG always outperforms FT-EMB in both settings.

E. NUMBER OF PROTOTYPES

In Section V-B, we determined the best number of prototypes on the validation set of the LS-100 task. Here we report the performance on LS- Test-Clean and LS-Test-Other of the LS-100 task using different numbers of prototypes. As shown in Fig. 12, WavLM performs best with 1000 prototypes, consistent with the results on the validation set, while HuBERT achieves the lowest error rates with 1500 prototypes; the results with 1000 prototypes are close behind, with only a slight difference between the two.

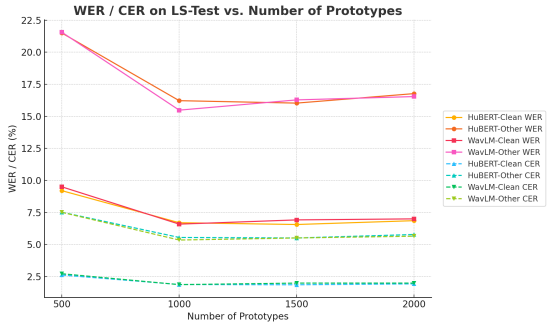


FIGURE 12: ASR performance of CATT-RPG with different numbers of prototypes tested on LS-Test-Clean and LS-Test-Other of the LS-100 task. The x-axis represents the number of prototypes, and the y-axis shows the WER (%) or CER (%) value.

F. CATT-RPG VOCABULARY SIZE ANALYSIS BEYOND DASB

As shown in Table 9, we compared the performance of CATT-RPG with different PLM vocabulary sizes (50260 and 6000) on ASR tasks other than DASB. Under low-resource conditions (i.e., LL-10 and Swedish), the 6000-token setting consistently outperforms the full vocabulary setting (50260). Interestingly, under rich-resource conditions (i.e., LS-960), the 6000-token configuration is still more effective than the full vocabulary setting, which is used in Time-LLM [9]. These findings suggest that our modified vocabulary design is a more appropriate and innovative adjustment for ASR tasks, possibly because the full vocabulary leads to more dispersed representations that are not as effectively adapted under PLM fine-tuning as the compact 6000-token set.

G. ANALYSIS OF ATTENTION WEIGHTS IN THE DSU EMBEDDING MODULE

We further analyzed the attention weight distribution of the six-layer MLP of the DSU embedding module on the Basque and Welsh test sets (trained on their respective training sets), as well as the LS-Test-Clean and LS-Test-Other sets (trained on LS-100 or LS-960), covering cross-lingual, low-resource, and rich-resource ASR settings. For English, as shown in Figs. 13(a) and 13(b) for LS-960 and 13(d) and 13(e) for LS-100, CATT-RPG consistently assigns higher weights to shallow SSL layers, in contrast to FT-EMB, which tends to emphasize deeper layers, revealing a fundamental difference between the two approaches in utilizing SSL representations. Previous research has shown that deeper layers of SSL models primarily encode semantic information, while shallower layers capture acoustic features [51]. Therefore, the fact that CATT-RPG relies more on shallower layers suggests that it prioritizes acoustic information in aligning codeword features with PLM semantics.

Basque also exhibits a similar trend (Fig. 13(c)), with higher weights in the shallow layers. However, the weight patterns of the intermediate layers (e.g., L7 and L18) differ between Basque and English. This difference suggests that while Basque exhibits a preference for shallow acoustic cues, similar to the trend observed in English tasks, CATT-RPG also relies on intermediate SSL layers that encode mixed acoustic–semantic information due to cross-lingual mismatches arising from applying English-trained SSL codeword extraction to Basque speech. In contrast, for the low-resource Welsh language (Fig. 13(f)), intermediate layers between the acoustic and semantic representations receive greater weights. These findings suggest that when given sufficient training data, CATT-RPG is able to automatically learn a mechanism to bridge acoustic and semantic information within PLM embeddings. In cross-lingual scenarios with abundant training resources, the model tends to extract as much information as possible from shallow layers, reflecting their greater contribution in capturing transferable acoustic patterns. However, under low-resource conditions, the limited amount of data makes it difficult to rely solely on acoustic features, so the mechanism shifts to intermediate layers that contain mixed semantic information.

Since WavLM uses data augmentation during pre-training, its features are inherently more robust than those of HuBERT. As shown in our experiments on Welsh and previous research [51], this augmentation particularly benefits acoustic information. As a result, WavLM achieves greater gains than HuBERT under low-resource and cross-lingual conditions, likely due to its greater emphasis on shallow layers rich in acoustic information.

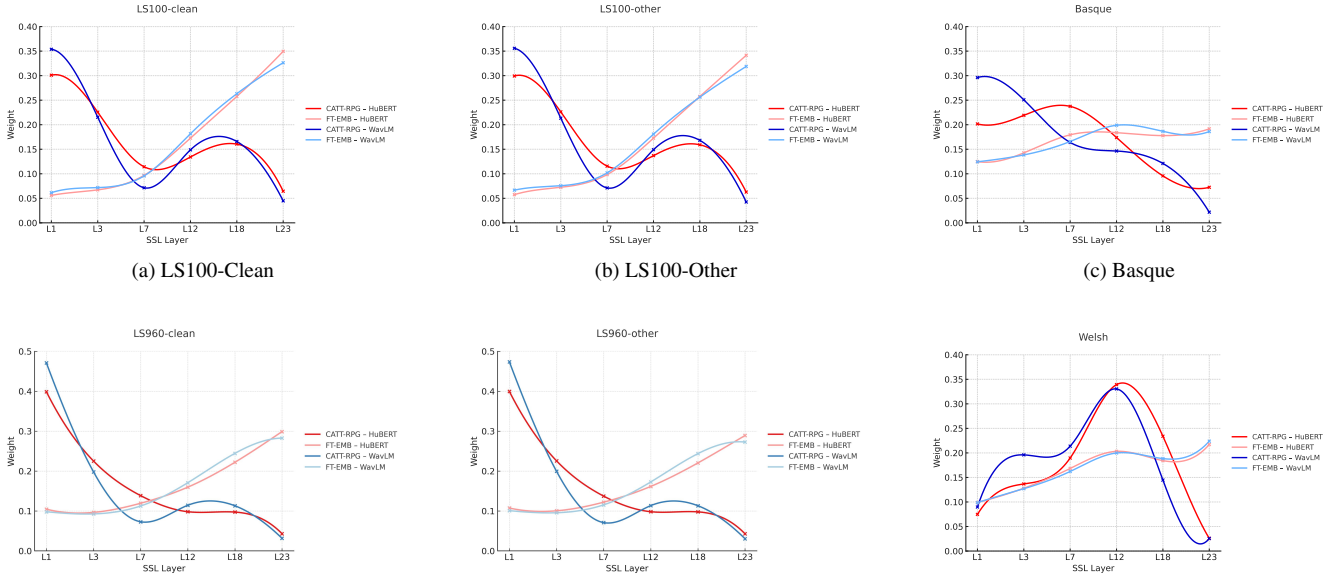
This observation, consistent with the findings in Table 7, further confirms the potential complementarity between CATT-RPG and direct DSU-based embeddings.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a cross-attention reprogramming method to improve DSU-based ASR. Our experiments show that the proposed CATT-RPG mechanism achieves SOTA performance in almost all DASB ASR settings, and consistently outperforms the baseline method of directly fine-tuning

TABLE 9: WER and CER results of CATT-RPG with different vocabulary sizes on LL-10, LS-960, and Swedish ASR tasks. Underlined: the best performance for each SSL Extractor.

SSL Extractor	CATT-RPG Vocab	WER					CER				
		LL-10		LS-960		Swedish	LL-10		LS-960		Swedish
		Clean	Other	Clean	Other		Clean	Other	Clean	Other	
HuBERT	6000	21.16	30.88	3.75	9.23	70.26	6.21	10.49	1.02	3.12	24.02
	50260	22.74	34.01	3.74	9.43	72.33	6.60	11.82	1.03	3.17	24.78
WavLM	6000	19.61	29.91	3.76	9.08	65.96	5.86	10.50	1.00	3.07	21.67
	50260	22.29	32.98	3.78	9.34	67.11	6.56	11.57	1.01	3.13	22.17



(a) LS100-Clean (b) LS100-Other (c) Basque

(d) LS960-Clean (e) LS960-Other (f) Welsh

FIGURE 13: First row: LS100-Clean (a), LS100-Other (b), and Basque (c); second row: LS960-Clean (a), LS960-Other (b), and Welsh (c). LS100-Clean and LS100-Other indicate training on LS-100 and evaluation on LS-Test-Clean and LS-Test-Other; LS960-Clean and LS960-Other indicate training on LS-960 and evaluation on LS-Test-Clean and LS-Test-Other. For Basque and Welsh, the models are trained and evaluated on their respective corpora. Each figure shows the average attention weight distribution across six SSL layers for CATT-RPG and FT-EMB.

the PLM embedding layer on LS-960 and LL-10. Notably, we find that reprogramming yields particularly significant improvements in low-resource training and cross-lingual ASR tasks. Our experiments on Swedish, Czech, and Hungarian ASR further confirm this observation.

Through the ablation study, we further show that reprogramming facilitates a substantially tighter integration with the PLM than approaches that merely project codeword features through a simple linear layer. Moreover, we observe that applying reprogramming after pooling DSU features across multiple SSL layers leads to better performance. Additionally, we validate the generalizability and effectiveness of the proposed method by comparing randomly selecting 6000 PLM token embeddings with selecting the first 6000 for reprogramming on the LS-100 ASR task. In summary, we demonstrate the benefits of bridging DSUs of an SSL speech model with a PLM via cross-attention-based reprogramming in ASR.

In future work, we aim to extend this framework to support generation-oriented architectures by integrating with large language models, replacing the current reliance on CTC and search. We also plan to perform joint training on multiple cross-lingual and low-resource languages and further improve the integration of DSUs and codewords to enhance the CATT-RPG module. Furthermore, we envisage integrating richer traditional acoustic features (e.g., MFCCs) to complement the SSL-based DSU generation process, using complete time-frequency representations (e.g., spectrograms) and phase information, which can capture cues related to speech intelligibility and speaker traits. These feature types can provide complementary, fine-grained acoustic details that SSL discretization cannot fully preserve. By utilizing these auxiliary acoustic cues, future extensions of this work are expected to improve robustness to noise, channel variations, and speaker differences, especially in low-resource and cross-lingual ASR scenarios.

Beyond ASR, we also seek to apply this reprogramming framework to spoken language understanding tasks, such as spoken question answering and speech emotion recognition. Given the promising advantages observed in low-resource and cross-lingual scenarios, we are particularly keen to experiment with more languages, including non-Latin ones such as Bulgarian. Finally, we acknowledge that the incorporation of reprogramming introduces additional parameters. Mitigating this computational overhead remains a pivotal direction for future work.

TABLE 10: Examples of FT-EMB and CATT-RPG ASR transcriptions across datasets and feature extractors.

Dataset	Example
LS100-Clean (HuBERT)	**Good:** *CATT-RPG:* COMBE THE WOOL FOR THE WIDOAWS INDEED *FT-EMB:* COME THE WOL FOR THE WEDAWRS IN DEED *Ref:* COMB THE WOOL FOR THE WHITTAWS INDEED **Bad:** *CATT-RPG:* AND MINE IS WILL STUTLEY SHALL WE BE COME RADS *FT-EMB:* AND MINE IS WILL STUTLEY SHALL WE BE COMRADES *Ref:* AND MINE IS WILL STUTELEY SHALL WE BE COMRADES
LS100-Clean (WavLM)	**Good:** *CATT-RPG:* THANK YOU DORCAS DEAR *FT-EMB:* THNK YOU DORKAST DEAR *Ref:* THANK YOU DORCAS DEAR **Bad:** *CATT-RPG:* THERE'S NOTHING ELSE THAT LOOKED SO JOLLY *FT-EMB:* THERE IS NOTHING ELSE THAT LOOK SO JOLLY *Ref:* THERE IS NOTHING ELSE THAT LOOKS SO JOLLY
LS100-Other (HuBERT)	**Good:** *CATT-RPG:* CHRIST LOOKED FROM A ICLEL PLAITED FLASHLIGHT TO A CAR JACK AND SPARK PLUG *FT-EMB:* CHRIST LOOK FROM I NICAL PLAITED FLASHLIGHT TO A CARRA JAK AND SPARK PLOOG *Ref:* CHRIS LOOKED FROM A NICKEL PLATED FLASHLIGHT TO A CAR JACK AND SPARK PLUG **Bad:** *CATT-RPG:* AL STRIGHTFUL TRIDTFUL *FT-EMB:* OH SFRIGHTFUL DRIDTFUL *Ref:* OH FRIGHTFUL FRIGHTFUL
LS100-Other (WavLM)	**Good:** *CATT-RPG:* SOME SUNG OUT TEAR DOWN THE FENCE TEAR DOWN THE FENCE *FT-EMB:* SOME SONG OUT TEARED DOWN THE FENDS TEAR DOWN THE FENTS *Ref:* SOME SUNG OUT TEAR DOWN THE FENCE TEAR DOWN THE FENCE **Bad:** *CATT-RPG:* HOW HAWR ISTILL INCINATIONS *FT-EMB:* HOW HAVE I TILL INCLINATIONS *Ref:* HOW HAVE I STILL INCLINATION
Basque (HuBERT)	**Good:** *CATT-RPG:* Baina zer da sinisgarritasuna zelofanezko mundu honetan? *FT-EMB:* Baina zer da sinisgarritasuna zelafanezko munduoenetan . *Ref:* Baina zer da sinesgarritasuna zelofanezko mundu honetan? **Bad:** *CATT-RPG:* Hainlo zatia bazara ezizu in oiz ez z lortuko? *FT-EMB:* Hainotzatia bazara ez duzu inoiz ezer lortuko. *Ref:* Hain lotsatia bazara ez duzu inoiz ezer lortuko.
Basque (WavLM)	**Good:** *CATT-RPG:* Baina zer da sinesgarritasuna, zelofanezko mundu honetan? *FT-EMB:* Baina zer da ezinisgarritasuna zelafenezko mundu honetan A. *Ref:* Baina zer da sinesgarritasuna zelofanezko mundu honetan? **Bad:** *CATT-RPG:* Elgiriak bostteikaraman zieden baina ez zirene benetanna eizkidetu. *FT-EMB:* Elkarri bosteka eman zideten, baina ez ziren benetan aiskidetu. *Ref:* Elkarri bostekoa eman zioten, baina ez ziren benetan adiskidetu.

APPENDIX A QUALITATIVE EXAMPLES OF CATT-RPG AND FT-EMB

To provide an intuitive understanding of how CATT-RPG differs from FT-EMB, we present representative transcription examples across English and Basque datasets under both HuBERT and WavLM extractors in Table 10. Each case includes both ''Good'' and ''Bad'' examples to illustrate situations where CATT-RPG improves over FT-EMB and where both methods may fail differently.

On LS100-Clean (HuBERT), CATT-RPG tends to maintain more accurate lexical forms (e.g., WHITTAWS $\rightarrow$ WIDOAWS) while FT-EMB produces phonetically similar but orthographically inconsistent outputs. In contrast, on LS100-Other, both models occasionally exhibit instability under rapid or noisy segments, though CATT-RPG generally preserves better temporal coherence. For WavLM-based decoding, CATT-RPG generates cleaner and more fluent sentences (e.g., THANK YOU DORCAS DEAR), while FT-EMB often suffers from partial omissions or morphological drift. In the Basque task, the difference becomes clearer: FT-EMB tends to fragment rare or long tokens into nonsensical forms (e.g., zelafenezko munduoenetan), whereas CATT-RPG reconstructs linguistically valid and semantically consistent sentences (e.g., zelofanezko mundu honetan). These examples collectively reveal complementary characteristics between the two methods—CATT-RPG emphasizes linguistic consistency and form accuracy, while FT-EMB often prioritizes acoustic smoothness at the cost of lexical precision.

ACKNOWLEDGMENTS

The authors thank Ming-Chi Yen for maintaining the GPU server infrastructure and Pooneh Mousavi et al. for providing the DASB dataset and related recipes. The authors also thank OpenAI for providing access to the GPT-4o language model and Google for access to the Gemini Flash 2.5 model, both of which were used solely to assist in polishing the language and enhancing the clarity and readability of the manuscript across all sections.

REFERENCES

[1] E. Kharitonov, J. Copet, K. Lakhotia, T. A. Nguyen, P. Tomasello, A. Lee, A. Elkahky, W.-N. Hsu, A. Mohamed, E. Dupoux, and Y. Adi, "Textless-lib: A library for textless spoken language processing," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol. (NAACL-HLT), Syst. Demonstrations*, Seattle, WA, USA (hybrid), Jul. 2022. [Online]. Available: https://aclanthology.org/2022.naacl-demo.1/

[2] A. Baevski, S. Schneider, and M. Auli, "VQ-Wav2vec: Self-supervised learning of discrete speech representations," in *Proc. 8th Int. Conf. Learn. Represent. (ICLR)*, Addis Ababa, Ethiopia, Apr. 2020.

[3] X. Chang, B. Yan, K. Choi, J.-W. Jung, Y. Lu, S. Maiti, R. Sharma, J. Shi, J. Tian, S. Watanabe, Y. Fujita, T. Maekaku, P. Guo, Y.-F. Cheng, P. Denisov, K. Saijo, and H.-H. Wang, "Exploring speech recognition, translation, and understanding with discrete speech units: A comparative study," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Seoul, Republic of Korea, Apr. 2024. [Online]. Available: https://doi.org/10.1109/ICASSP48485.2024.10447929

[4] X. Chang, J. Shi, J. Tian, Y. Wu, Y. Tang, Y. Wu, S. Watanabe, Y. Adi, X. Chen, and Q. Jin, "The Interspeech 2024 challenge on speech processing using discrete units," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Dublin, Ireland, Sep. 2024.

[5] W. Nakata, K. Yamauchi, D. Yang, H. Hyodo, and Y. Saito, "UTDUSS: UTokyo-SaruLab system for Interspeech2024 speech processing using discrete speech unit challenge," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Sept. 2024.

[6] G.-T. Lin, Y.-S. Chuang, H.-L. Chung, S.-w. Yang, H.-J. Chen, S. Dong, S.-W. Li, A. Mohamed, H.-y. Lee, and L.-s. Lee, "DUAL: Discrete spoken unit adaptive learning for textless spoken question answering," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Incheon, Republic of Korea, Sept. 2022.

[7] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9054224

[8] J. Shi, C.-J. Hsu, H. Chung, D. Gao, P. Garcia, S. Watanabe, A. Lee, and H.-Y. Lee, "Bridging speech and textual pre-trained models with unsupervised ASR," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023, pp. 1–5.

[9] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan, and Q. Wen, "Time-LLM: Time series forecasting by reprogramming large language models," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, Apr. 2024.

[10] Y. Wang, H. Wu, J. Dong, Y. Liu, M. Long, and J. Wang, "Deep time series models: A comprehensive survey and benchmark," 2024, *arXiv:2407.13278*.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NeurIPS)*, Long Beach, CA, USA, Dec. 2017.

[12] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, A. Rodriguez, A. Joulin, E. Grave, and G. Lample, "LLaMA: Open and efficient foundation language models," 2023, *arXiv:2302.13971*.

[13] P. Mousavi, L. Della Libera, J. Duret, A. Ploujnikov, C. Subakan, and M. Ravanelli, "DASB – Discrete audio and speech benchmark," 2024, arXiv:2406.14294.

[14] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, T.-H. Huang, W.-C. Tseng, K.-t. Lee, D.-R. Liu, Z. Huang, S. Dong, S.-W. Li, S. Watanabe, A. Mohamed, and H.-y. Lee, "SUPERB: Speech processing universal performance benchmark," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Brno, Czech Republic, Aug. 2021, pp. 1194–1198. [Online]. Available: https://doi.org/10.21437/Interspeech.2021-1775

[15] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, South Brisbane, QLD, Australia, Apr. 2015. [Online]. Available: https://doi.org/10.1109/ICASSP.2015.7178964

[16] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux, "Libri-Light: A benchmark for ASR with limited or no supervision," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Barcelona, Spain, May 2020. [Online]. Available: https://doi.org/10.1109/ICASSP40776.2020.9052942

[17] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, "Common Voice: A massively-multilingual speech corpus," in *Proc. 12th Lang. Res. Eval. Conf. (LREC)*, Marseille, France, May 2020. [Online]. Available: https://aclanthology.org/2020.lrec-1.520/

[18] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and E. Dupoux, "On generative spoken language modeling from raw audio," *Trans. Assoc. Comput. Linguist.*, vol. 9, pp. 1336–1354, Feb. 2021.

[19] Z. Borsos, R. Marinier, D. Vincent, E. Kharitonov, O. Pietquin, M. Sharifi, D. Roblek, O. Teboul, D. Grangier, M. Tagliasacchi, and N. Zeghidour, "AudioLM: A language modeling approach to audio generation," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 31, pp. 1234–1245, Jan. 2023.

[20] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, Oct. 2021.

[21] S. Chen, Y. Wu, C. Wang, S. Liu, D. Tompkins, Z. Chen, W. Che, X. Yu, and F. Wei, "BEATs: Audio pre-training with acoustic tokenizers," in *Proc. 40th Int. Conf. Mach. Learn. (ICML)*, Honolulu, HI, USA, Jul. 2023. [Online]. Available: https://proceedings.mlr.press/v202/chen23ag.html

[22] F. Wu, K. Kim, S. Watanabe, K. J. Han, R. McDonald, K. Q. Weinberger, and Y. Artzi, "Wav2Seq: Pre-training speech-to-text encoder-decoder models using pseudo languages," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023. [Online]. Available: https://doi.org/10.1109/ICASSP49357.2023.10096988

[23] X. Chang, B. Yan, Y. Fujita, T. Maekaku, and S. Watanabe, "Exploration of efficient end-to-end ASR using discretized input from self-supervised learning," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Dublin, Ireland, Aug. 2023. [Online]. Available: https://doi.org/10.21437/Interspeech.2023-2051

[24] L.-W. Chen, S. Watanabe, and A. Rudnicky, "A unified one-shot prosody and speaker conversion system with self-supervised discrete speech units," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Rhodes Island, Greece, Jun. 2023. [Online]. Available: https://doi.org/10.1109/ICASSP49357.2023.10095565

[25] X. Cheng, Z. Zhu, Z. Yao, H. Li, Y. Li, and Y. Zou, "GhostT5: Generate more features with cheap operations to improve textless spoken question answering," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Dublin, Ireland, Aug. 2023. [Online]. Available: https://doi.org/10.21437/Interspeech.2023-41

[26] M. Cui, D. Tan, Y. Yang, D. Wang, H. Wang, X. Chen, X. Chen, and X. Liu, "Exploring SSL discrete tokens for multilingual ASR," 2024, arXiv:2409.08805.

[27] H. Inaguma, S. Popuri, I. Kulikov, P.-J. Chen, C. Wang, Y.-A. Chung, Y. Tang, A. Lee, S. Watanabe, and J. Pino, "UnitY: Two-pass direct speech-to-speech translation with discrete units," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Toronto, Canada, Jul. 2023, pp. 15655–15680. [Online]. Available: https://doi.org/10.18653/v1/2023.acl-long.872

[28] S. Kando, Y. Miyao, and S. Takamichi, "Exploring the effect of segmentation and vocabulary size on speech tokenization for speech language models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Rotterdam, The Netherlands, Aug. 17–21, 2025, pp. 5728–5732.

[29] K. Onda, Y. Kashiwagi, E. Tsunoo, H. Futami, and S. Watanabe, "Differentiable K-means for fully-optimized discrete token-based ASR," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Rotterdam, The Netherlands, Aug. 17–21, 2025, pp. 10401–10405.

[30] J. Shi, X. Ma, H. Inaguma, A. Sun, and S. Watanabe, "MMM: Multi-layer multi-residual multi-stream discrete speech representation from self-supervised learning model," in *Proc. Annu. Conf. Int. Speech Commun.*

Assoc. (Interspeech), Dublin, Ireland, Sept. 2024, pp. 2569–2573. [Online]. Available: https://doi.org/10.21437/Interspeech.2024-2251

[31] S. Shon, K. Kim, Y.-T. Hsu, P. Sridhar, S. Watanabe, and K. Livescu, "DiscreteSLU: A large language model with self-supervised discrete speech units for spoken language understanding," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Incheon, South Korea, Sept. 2024.

[32] S.-H. Wang, J. Shi, C.-Y. Huang, S. Watanabe, and H.-Y. Lee, "Fusion of discrete representations and self-augmented representations for multilingual automatic speech recognition," in *Proc. IEEE Spoken Lang. Technol. Workshop (SLT)*, Doha, Qatar, Jan. 2024, pp. 247–254.

[33] J. Zhao, N. Moritz, E. Lakomkin, R. Xie, Z. Xiu, K. Zmolikova, Z. Ahmed, Y. Gaur, D. Le, and C. Fuegen, "Textless streaming speech-to-speech translation using semantic speech tokens," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Hyderabad, India, Apr. 2025, pp. 1–5. [Online]. Available: https://doi.org/10.1109/ICASSP49660.2025.10889740

[34] T. T. Dieck, P. A. Pérez-Toro, T. Arias, E. Noeth, and P. Klumpp, "Wav2vec behind the scenes: How end-to-end models learn phonetics," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Incheon, Korea, Sept. 2022, pp. 5130–5134. [Online]. Available: https://doi.org/10.21437/Interspeech.2022-10865

[35] M. Mohamed, O. D. Liu, H. Tang, and S. Goldwater, "Orthogonality and isotropy of speaker and phonetic information in self-supervised speech representations," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Kos Island, Greece, Sept. 2024. [Online]. Available: https://doi.org/10.21437/Interspeech.2024-1054

[36] T. Nakamura, K. Choi, K. Hojo, Y. Bando, S. Fukayama, and S. Watanabe, "Discrete speech unit extraction via independent component analysis," in *Proc. ICASSP 2025 Workshop on Self-supervised Learning for Audio, Music, and Speech Applications (SALMA)*, Hyderabad, India, Apr. 2025. [Online]. Available: https://doi.org/10.1109/ICASSPW65056.2025.11011236

[37] A. Maćkiewicz and W. Ratajczak, "Principal components analysis (PCA)," *Comput. Geosci.*, vol. 19, no. 3, pp. 303–308, May 1993.

[38] A. Hyvärinen and E. Oja, "Independent component analysis: Algorithms and applications," *Neural Netw.*, vol. 13, no. 4–5, pp. 411–430, Jun. 2000.

[39] S. Shon, S. Arora, C.-J. Lin, A. Pasad, F. Wu, R. Sharma, W.-L. Wu, H.-Y. Lee, K. Livescu, and S. Watanabe, "SLUE Phase-2: A benchmark suite of diverse spoken language understanding tasks," in *Proc. 61st Annu. Meeting Assoc. Comput. Linguist. (ACL)*, Toronto, Canada, Jul. 2023, pp. 8906–8937. [Online]. Available: https://doi.org/10.18653/v1/2023.acl-long.496

[40] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn. (ICML)*, Pittsburgh, PA, USA, Jun. 2006, pp. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[41] A. Amjad, L.-C. Tai, and H.-T. Chang, "Utilizing enhanced particle swarm optimization for feature selection in gender-emotion detection from English speech signals," IEEE Access, vol. 12, pp. 189564–189573, 2024, doi: 10.1109/ACCESS.2024.3516790.

[42] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE J. Sel. Topics Signal Process.*, vol. 16, no. 6, pp. 1505–1518, Oct. 2022.

[43] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Bidirectional LSTM networks for improved phoneme classification and recognition," in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*, Warsaw, Poland, Sep. 2005, pp. 799–804.

[44] S. Na, L. Xumin, and G. Yong, "Research on K-means clustering algorithm: An improved K-means clustering algorithm," in *Proc. 3rd Int. Symp. Intell. Inf. Technol. Security Informat. (IITSI)*, Jian, China, Apr. 2010.

[45] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," 2020, arXiv:2004.05150.

[46] Y. Sudo, M. Shakeel, Y. Fukumoto, B. Yan, J. Shi, Y. Peng, and S. Watanabe, "Joint beam search integrating CTC, attention, and transducer decoders," 2024, *arXiv:2406.02950*.

[47] S.-C. Chiu, C.-H. Wu, J.-K. Hsieh, Y. Tsao, and H.-M. Wang, "Learnable layer selection and model fusion for speech self-supervised learning models," in *Proc. Annu. Conf. Int. Speech Commun. Assoc. (Interspeech)*, Sep. 2024, pp. 3914–3918. [Online]. Available: https://doi.org/10.21437/Interspeech.2024-849

[48] P.-J. Liao, H.-Y. Lee, and H.-M. Wang, "Ensemble knowledge distillation from speech SSL models considering inter-teacher differences," in *Proc. 14th IEEE Int. Symp. Chinese Spoken Lang. Process. (ISCSLP)*, Beijing, China, Nov. 2024, pp. 716–720. [Online]. Available: https://doi.org/10.1109/ISCSLP63861.2024.10800400

[49] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, and others, "Transformers: State-of-the-art natural language processing," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP) – System Demonstrations*, Online, Nov. 2020, pp. 38–45. [Online]. Available: https://aclanthology.org/2020.emnlp-demos.6

[50] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. De Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021, arXiv:2106.04624.

[51] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Rhodes, Greece, Jun. 2023, pp. 1–5.

PEI-JUN LIAO received the B.S. degree from the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, and the M.S. degree from the Graduate Institute of Communication Engineering, National Taiwan University. He is currently pursuing a Ph.D. degree at the Graduate Institute of Electrical Engineering, National Taiwan University. His research interests include speech processing and self-supervised learning.

HUNG-YI LEE (Senior Member, IEEE) is a professor of EE and CS at National Taiwan University (NTU). His recent research focuses on developing technology that can reduce the requirement of annotated data for speech processing and natural language processing. He is a co-organizer of the workshop "Self-Supervised Learning for Speech and Audio Processing" at NeurIPS (2020) and AAAI (2022), special issue "New Trends in self-supervised speech processing" at INTERSPEECH (2020), workshop "Meta Learning and Its Applications to Natural Language Processing" at ACL (2021). He was the technical program co-chair of ISCSLP 2022, technical program co-chair of INLG 2023 and program chair of Machine Learning Summer School (MLSS) 2021 Taipei. He was a lead guest editor of the Journal of Selected Topics in Signal Processing (JSTSP), editing a special issue, "Self-Supervised Learning for Speech and Audio Processing".

HSIN-MIN WANG (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from National Taiwan University, Taipei, Taiwan, in 1989 and 1995, respectively. In October 1995, he joined the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Research Fellow. He was a Joint Professor in the Department of Computer Science and Information Engineering at National Cheng Kung University from 2014 to 2023. He was an Associate Editor of IEEE/ACM Transactions on Audio, Speech and Language Processing from 2016 to 2020. He currently serves a Senior Editor of APSIPA Transactions on Signal and Information Processing. His major research interests include spoken language processing, natural language processing, multimedia information retrieval, and machine learning. He was a General Co-Chair of ISCSLP2016, ISCSLP2018, ASRU2023, and O-COCOSDA2024 and a Technical Co-Chair of ISCSLP2010, O-COCOSDA2011, APSIPAASC2013, ISMIR2014, ASRU2019, and APSIPAASC2023. He received the Chinese Institute of Engineers Technical Paper Award in 1995 and the ACM Multimedia Grand Challenge First Prize in 2012. He was an APSIPA distinguished lecturer for 2014–2015. He is a member of ISCA, ACM, and APSIPA.