Latent-RQ: Enhancing Speech Pre-Training with Latent Representations and Random Quantization

Muhammad Huzaifah HUZAIFAH_MD_SHAHRIN@A-STAR.EDU.SG
Hardik B. Sailor SAILOR_HARDIK_BHUPENDRA@A-STAR.EDU.SG
Jeremy H. M. Wong JEREMY_WONG@A-STAR.EDU.SG
Nancy F. Chen NANCY_CHEN@A-STAR.EDU.SG
Ai Ti Aw AW_AI_TI@A-STAR.EDU.SG
*Institute for Infocomm Research (I²R), Agency for Science, Technology and Research (A*STAR), Singapore*

Editors: Tatsuya Komatsu, Keisuke Imoto, Xiaoxue Gao, Nobutaka Ono, Nancy F. Chen

Abstract

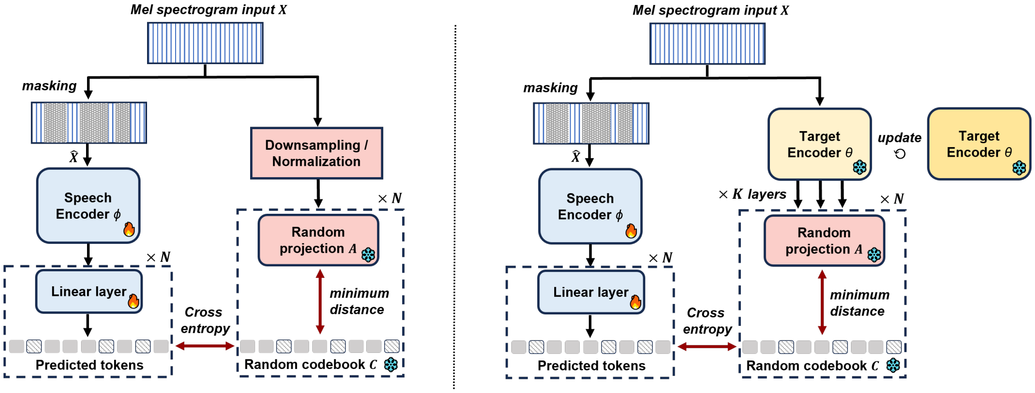
Random quantization is a simple yet effective strategy for speech self-supervised pre-training, producing strong encoder representations for a range of downstream tasks. However, existing methods such as BEST-RQ rely on Mel spectrograms – low-level acoustic features – as quantizer inputs, which may hinder convergence and limit target quality. We propose Latent-RQ, an extension that replaces direct Mel inputs with richer latent representations extracted from a pre-trained encoder. To further enhance target quality, the target encoder is periodically updated during training. Latent-RQ achieves consistent improvements on the SUPERB benchmark, particularly for speech recognition and speaker identification, while reaching comparable performance to BEST-RQ with fewer optimization steps under a fixed training budget. We also analyze how target layer selection influences downstream performance and layer-wise information encoding. t-SNE visualizations of phoneme and speaker embeddings reveal clearer clustering and improved target discriminability. Overall, Latent-RQ offers a scalable and effective enhancement to random quantization–based SSL frameworks for speech representation learning.

Keywords: Latent-RQ, self-supervised learning, BEST-RQ, speech representations, pre-training, latent features

1. Introduction

Self-supervised learning (SSL) has emerged as a transformative approach in speech processing, enabling models to learn meaningful representations from large-scale unlabeled audio data (Bengio et al., 2013). By pre-training on proxy tasks designed to uncover the latent structure in speech, SSL models can generalize across a variety of downstream applications such as automatic speech recognition (ASR), speaker identification (SID), and emotion recognition (ER) (Mohamed et al., 2022). Among SSL approaches, a common predictive strategy adopted by HuBERT (Hsu et al., 2021), WavLM (Chen et al., 2021), and BEST-RQ (Chiu et al., 2022) among others (Baevski et al., 2019; Huang et al., 2025; Chen et al., 2024; Baevski et al., 2022), involves masked prediction, where the model is trained to infer masked portions of the input based on surrounding context (Joshi et al., 2020).

A key component of this framework is the design of the prediction targets, which provide the learning signal for masked positions. For current state-of-the-art SSL methods, targets



(a) BEST-RQ framework with multiple codebooks

(b) Latent-RQ framework with multiple layer targets and codebooks

Figure 1: (a) The original BEST-RQ framework. Components that are trained are depicted with a fire symbol while those that are frozen are depicted with a snowflake. (b) Our proposed Latent-RQ framework inserts an additional encoder into the target creation pipeline which feeds the random quantizer with learned features instead of the Mel spectrogram directly. The target encoder could in theory be any sufficiently trained speech encoder model, although in this work we initialize it with the weights of the speech encoder ϕ itself and replace it periodically with an updated snapshot as training progresses.

are typically discrete tokens transformed from continuous speech, which have been shown to be more performant and easier to train than the latter (Nguyen et al., 2022). For example, in HuBERT and WavLM, targets are obtained via iterative k-means clustering of intermediate hidden states or acoustic features like Mel-frequency cepstral coefficients (MFCCs), with the cluster assignments serving as pseudo-labels. The quality and nature of these targets heavily influence the downstream utility of the learned representations. For instance, coarse or unstable clustering may limit phonetic precision, while overly granular targets may yield noisy gradients and reduce generalizability (Chen et al., 2025; Hsu et al., 2021).

BEST-RQ (BERT-based Speech Pre-training with Random-projection Quantizer) offers a compelling lightweight alternative by replacing iteratively computed cluster assignments with a fixed, random-projection-based quantizer. Instead of the computationally expensive k-means clustering, this method projects high-dimensional input features (typically Mel spectrograms) into a lower-dimensional space using a randomly initialized matrix, followed by nearest-neighbor quantization through a static, randomized codebook that is not learned. This design eliminates the need for multiple rounds of offline clustering or learned quantizers, yielding a more streamlined, scalable, and possibly more compute-efficient training pipeline. Additionally, the fixed nature of the quantizer ensures target stability and avoids codebook collapse, which can be problematic in fully end-to-end learned systems (Chung et al., 2021).

Despite its simplicity and strong performance, BEST-RQ presents a key limitation: it applies random quantization directly to Mel spectrograms, which are relatively low-level representations that may not adequately express higher-order acoustic or linguistic structures. The random projection primarily serves as a form of dimensionality reduction, approximately preserving pairwise distances in the feature space as described by the Johnson–Lindenstrauss lemma (Johnson and Lindenstrauss, 1984), but does not inherently enhance the representational richness of the features. As a result, the quantizer may require more training time to learn complex acoustic and semantic relationships that are critical for downstream tasks. The decoupling of the quantizer from a jointly learned representation pipeline restricts the expressiveness of the resulting targets and may weaken the overall learning signal. Empirically, it has been observed that the BEST-RQ approach converges more slowly than methods with alternative target design (Chiu et al., 2022; Chen et al., 2025; Huzaifah et al., 2024).

In this work, we propose *Latent-RQ* to address these limitations by leveraging learned latent representations as input to the random projection quantizer instead of raw Mel spectrograms. By injecting semantic abstraction from learned layers of a trained target encoder into the quantization stage, we enable the quantizer to operate on features that are more aligned with phonetic and prosodic structure, leading to targets that better reflect the underlying variability and patterns in speech. While the use of targets derived from latent features is in itself not new, as seen in HuBERT and WavLM, this is the first time it is combined with a frozen random projection target pipeline. This preserves architectural simplicity and scalability of the latter, without necessitating the offline clustering used in HuBERT or WavLM, while significantly improving the informativeness and task relevance of the targets.

This paper presents our initial exploration of incorporating latent representations into the random projection–based SSL framework, with the primary goal of benchmarking performance against an equivalent BEST-RQ model. We hypothesize that enriching the quantizer input with latent features yields more transferable representations, that would lead to faster convergence by requiring fewer optimization steps to reach a certain performance level. To verify this, we periodically evaluate both models on a suite of downstream tasks throughout training. For a fair comparison, we fix the total training budget to a predefined number of optimization steps. Experimental results demonstrate that Latent-RQ achieves comparable performance to BEST-RQ in fewer steps, particularly for ASR and SID tasks, while results for intent classification (IC) and emotion recognition (ER) are more mixed. Furthermore, regularly updating the target latent representations during training amplifies this advantage up to a point of saturation. We also conduct detailed analyses of how different layer combinations used for target generation influence downstream performance. In addition, t-SNE (Van der Maaten and Hinton, 2008) visualizations are employed to compare the clustering behavior of latent features against those derived from Mel spectrograms, providing deeper insights into target quality. Overall, our findings indicate that augmenting quantizer inputs with learned representations offers a principled direction for improving target design in speech SSL, combining the robustness and simplicity of random quantization with the expressive power of learned features.

2. Method

2.1. Masked Speech Prediction

Following BEST-RQ (Chiu et al., 2022), we adopt a masked prediction framework inspired by masked language modeling (Devlin et al., 2019). The SSL objective involves predicting discrete targets corresponding to masked segments of a speech signal, as illustrated in Fig. 1. Discrete targets are drawn from multiple independent codebooks, a design choice that improves training stability and representation capacity (Zhang et al., 2023).

Let $X = [x_1, x_2, \ldots, x_T]$ denote an input sequence of T log-Mel spectrogram frames and $\mathcal{C} = \{C_1, C_2, \ldots, C_N\}$ be a set of N random codebooks, where each codebook $C_n \in \mathbb{R}^{V \times D}$ contains V codewords of dimension D. Each codebook is associated with a fixed, randomly initialized projection matrix $\mathbf{A}_n \in \mathbb{R}^{D' \times D}$ that projects input features $x_t \in \mathbb{R}^{D'}$ into the codebook space. Before projection, the input X is downsampled to match the temporal resolution of the encoder output, obtaining a frame sequence indexed by t. For each codebook n, the projected feature $\mathbf{A}_n x_t$ is compared to all codewords in C_n using Euclidean distance to obtain a discrete pseudo-label y

$$y_{t,n} = \arg\min_{v} \|c_{v,n} - \mathbf{A}_n x_t\|_2 \tag{1}$$

where $c_{v,n}$ denotes the v-th codeword in codebook C_n.

To construct the learning signal, we apply span-masking to the input sequence with a fixed probability, replacing a span of M frames with a learned mask embedding $\hat{x}_m$. A speech encoder ϕ maps the corrupted input $\hat{X}$ to a sequence of contextual hidden representations, which are projected into per-codebook distributions. The model is optimized via a cross-entropy loss computed over masked time steps and all codebooks

$$L = \sum_{n=1}^{N} \sum_{t \in \mathcal{M}} \log p_{\phi}(y_{t,n} \mid \hat{X}) \tag{2}$$

where $\mathcal{M}$ denotes the set of masked time steps, $y_{t,n}$ are the reference targets from the random quantizer and p_{ϕ} is the predicted distribution over codewords for each codebook.

2.2. Random Quantization with Latent Representations

Rather than deriving discrete targets directly from Mel spectrogram features, we propose to extract targets from a more abstract latent representation produced by a pre-trained speech encoder. This step is similar to the initial target extraction in WavLM (derived from the HuBERT model) and the second iteration of HuBERT (derived from its first iteration after initially training on clustered MFCC features). GS-16, an independent attempt to reproduce HuBERT (Chen et al., 2023), also proposes to derive initial targets from a trained ASR model.

Specifically, we introduce a frozen *target encoder* θ, initialized from the weights of the main speech encoder ϕ after initially training ϕ for a certain number of steps with the baseline BEST-RQ setup i.e. directly with Mel-spectrogram inputs. The target encoder produces contextualized features that serve as input to the random quantizer, potentially yielding more informative and structured target distributions.

Although any sufficiently-trained encoder may be used to derive targets in principle, employing the same architecture for both ϕ and θ has certain advantages. Namely, it aligns the temporal resolution of input features and targets, eliminating the need for the additional downsampling steps used previously. Let $\{h_k(X)\}_{k=1}^{K}$ denote the latent representations extracted from K intermediate layers of θ. Each $h_k(X) \in \mathbb{R}^{D_k}$ is then projected via a layer-specific random matrix $\mathbf{A}_{n,k} \in \mathbb{R}^{D_k \times D}$ and quantized using its assigned codebook $C_{n,k}$. Each layer is assigned N/K codebooks. The revised quantization equation becomes

$$y_{t,n,k} = \arg\min_{v} \|c_{v,n,k} - \mathbf{A}_{n,k} h_{k,t}(X)\|_2 \tag{3}$$

where $c_{v,n,k}$ is the v-th codeword in codebook $C_{n,k}$ associated with layer k.

2.3. Regular Refinement of Targets

As the speech encoder ϕ continues to learn and improve over the course of training, we hypothesize that its representations become increasingly suitable as sources for target supervision (Xu et al., 2020). To exploit this, we periodically refresh the target encoder θ by replacing it with a snapshot of the current speech encoder ϕ, thereby refining the target representations in tandem with the model's progression. This approach differs fundamentally from the iterative clustering procedure in HuBERT and WavLM, which requires running offline k-means clustering on the extracted latent representations for the entire dataset to generate new target labels during each update. In contrast, our refinement process simply involves swapping θ with a more up-to-date copy of the weights of ϕ. Since the random projections and codebooks remain fixed throughout training, the updated encoder seamlessly produces new latent targets, while maintaining consistency in the quantization space. Softmax layers however are randomly re-initialized to cater to the new targets.

We refer to the interval between two such target updates as a *stage*. During each stage, the target encoder θ remains frozen, ensuring target stability and avoiding the training instabilities often observed with dynamic supervision such as gradient norm spikes or codebook collapse. Empirically, we observe stable convergence behavior across all stages.

3. Experimental Details

We adopt the Conformer architecture (Gulati et al., 2020) as the backbone for the speech encoder. The model comprises 17 Conformer layers with an embedding dimension of 512 and a feedforward dimension of 2048, resulting in approximately 150M parameters. A two-layer convolutional feature extractor precedes the Conformer blocks, applying a temporal downsampling factor of 4× to the input. As input features, we use 80-bin log-Mel spectrograms computed with a 25ms window and 10ms hop size. Following prior work (Huzaifah et al., 2024; Whetten et al., 2024), which demonstrated the benefits of increased masking probabilities compared to the original BEST-RQ setup (Chiu et al., 2022), we set the masking probability to 0.4 throughout pre-training.

Pre-training was conducted on the 960hr LibriSpeech corpus (Panayotov et al., 2015). As shown in Fig. 2, each stage begins with the target encoder re-initialized from the trained encoder of the preceding stage. To ensure fair comparison, all stages were trained to a total of 500k steps, inclusive of prior steps from earlier stages. This schedule allows us to measure the effect of latent target refinement on convergence and final performance under a fixed training budget. Stage 0 represents the baseline BEST-RQ setup where Mel spectrograms are directly quantized via the random projection quantizer. In subsequent stages, latent targets are extracted from the current target encoder and passed to the quantizer. Similar

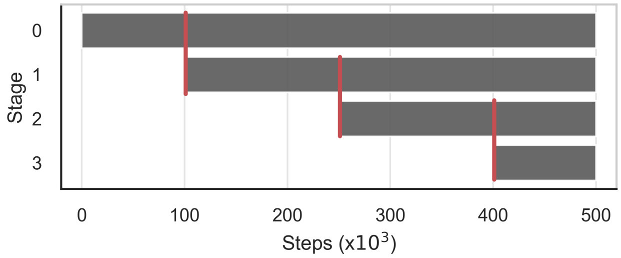


Figure 2: Training schedule across Latent-RQ stages. Stage 0 uses targets derived directly from Mel spectrograms. In subsequent stages, inputs are processed by a target encoder initialized from the previous stage. Specifically, stage 1 was initialized from the stage 0 encoder at 100k steps, stage 2 from stage 1 at 250k steps, and stage 3 from stage 2 at 400k steps (transitions denoted by red lines). Each stage continues training to a cumulative 500k steps, allowing direct comparison of performance across stages.

to HuBERT and WavLM, optimizer state, including the learning rate scheduler, is reset upon each target encoder update to encourage independent adaptation. Transition points occur at 100k, 250k, and 400k steps, yielding three total updates. We employ 18 independent codebooks for quantization, evenly distributed across the target layers used. Each codebook has a vocabulary size of 4096 and a dimension of 16. All runs were carried out on four H100 NVIDIA GPUs with a gradient accumulation of four.

4. Results

4.1. Downstream Performance of Latent-RQ

We evaluated Latent-RQ across multiple stages of training on the SUPERB benchmark (Yang et al., 2021), which measures the generalizability of speech representations across a suite of downstream tasks. To capture a diverse set of capabilities, we selected four representative tasks spanning different categories: automatic speech recognition (ASR), speaker identification (SID), intent classification (IC), and emotion recognition (ER). Following standard SUPERB protocol, embeddings were extracted from a frozen speech encoder and passed to the default downstream models with pre-defined training configurations. During pre-training, we derived quantization targets from intermediate layers 5 through 10 (zero-indexed) of the speech encoder, resulting in six latent layers used per stage. Performance across training steps and stages is shown in Fig. 3. The final results at 500k steps for Latent-RQ (stage 3) and BEST-RQ (stage 0) is given in Table 1 along with other state-of-the-art SSL models.

ASR and SID tasks show the clearest benefit from the Latent-RQ framework. Even a single application of latent target quantization (stage 1 vs. stage 0) yielded noticeable gains, while deeper stages (2 and 3) further enhanced performance, reflected in sharper improvements in word error rate (WER) for ASR and classification accuracy for SID. Gains were especially pronounced in SID, where stage 3 achieved nearly a 10% absolute improvement in

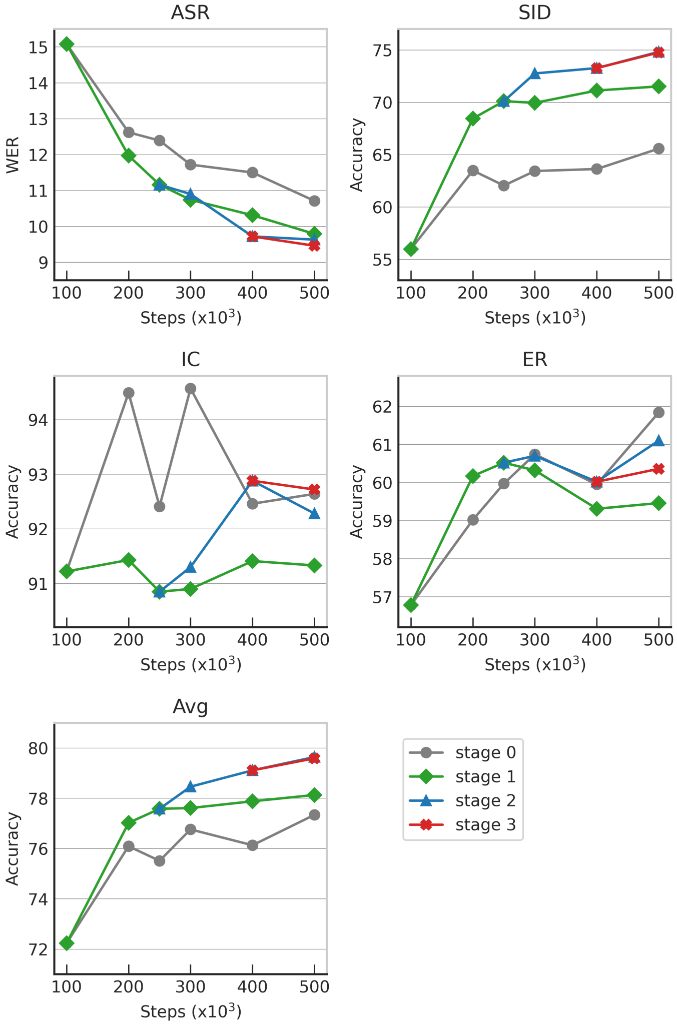


Figure 3: Performance trends over 500k steps for ASR, SID, IC, and ER tasks across various stages of training. The performance measure for ASR is word error rate (WER %) while SID, IC, and ER tasks use accuracy (%). The average (Avg) accuracy was calculated by subtracting WER from 100, before summing and dividing the accuracies from all four tasks.

accuracy over the baseline. Nevertheless, improvements seemingly saturate with increasing stages, given the minimal effect of stage 3 over stage 2.

In contrast, improvements were less consistent for IC and ER. For IC, accuracy fluctuated across stages and training steps, with the overall best result obtained by the stage 0 model at 300k steps. Similarly, in ER, the highest accuracy at 500k steps was achieved by stage 0, followed by stages 2, 3, and 1 in descending order. These outcomes suggest that latent quantization targets offer limited advantage, or in some cases, slight degradation, for certain tasks that may rely on information not effectively captured by the intermediate-layer

Table 1: Final results for Latent-RQ at stage 3 and the baseline BEST-RQ model (equivalent to stage 0) trained to 500k steps, against leading SSL models: Wav2vec2 Base (Baevski et al., 2020), HuBERT Base (Hsu et al., 2021) and WavLM Base (Chen et al., 2021).

Model	ASR	SID	ER	IC	Avg
Latent-RQ (Stage 3)	9.46	74.75	60.36	92.72	79.59
BEST-RQ (Stage 0)	10.71	65.57	61.84	92.64	77.34
Wav2vec2 Base	6.43	75.18	63.43	92.35	81.13
HuBERT Base	6.42	81.42	64.92	98.34	84.57
WavLM Base	6.21	84.51	65.94	98.63	85.72

representations used for target generation. This observation raises two important considerations. First, the latent representations selected for quantization may not be universally optimal across all task types. Tasks like IC and ER may require supervision signals derived from deeper or more task-aligned layers. We explore this hypothesis further in the following section. Second, the timing of the encoder snapshot used to initialize the target encoder may influence downstream performance. For instance, initializing stage 1 with the 100k checkpoint of the base model may not yield as strong a target representation for IC as a more mature checkpoint, such as at 200k steps. These findings highlight the sensitivity of latent target quality to both layer selection and update timing.

Despite these task-specific differences, Latent-RQ consistently improved overall performance on the SUPERB benchmark. As illustrated in the average score (Avg)[1] plot of Fig. 3, aggregated results improved at a faster rate for the same number of optimization steps with each stage, albeit with diminishing returns by stage 3. In our controlled experiment, Latent-RQ outperformed the vanilla BEST-RQ on three out of four tasks at 500k steps (Table 1), thus validating our central hypothesis that leveraging learned latent features can produce more effective targets compared to using Mel spectrograms alone. Nevertheless, the final result still underperforms current state-of-the-art SSL models given that the Latent-RQ model is still relatively undertrained at the current number of steps and GPU hours. We leave the scaling of the model training and a more complete comparison to other SSL models to future work.

4.2. Convergence and Throughput Trade-off

While Latent-RQ achieves comparable performance to BEST-RQ with substantially fewer optimization steps, it incurs a lower training throughput due to the additional forward passes required through the target encoder. We quantify this trade-off by comparing the reduction in throughput on our system against the decrease in the number of steps needed to reach equivalent performance. For simplicity, we report results based on the average accuracy shown in Fig. 3. With a gradient accumulation of four, each optimization step

[1] Average score was calculated using the same method popularized by WavLM (Chen et al., 2021), subtracting WER from 100 before averaging.

entails four extra forward passes through the target encoder relative to the baseline. As the target encoder is frozen, no gradient computation or parameter updates are performed. This reduces the average throughput from 0.445s to 0.570s per step. Nevertheless, since Latent-RQ attains the same performance level as BEST-RQ (at 500k steps) after only approximately 230k steps, this corresponds to a significant estimated time saving of about 25.4 hours, or over 40% of the total training time of BEST-RQ. Although this estimation may vary across training configurations, it strongly indicates the potential for substantial reductions in GPU hours using the proposed approach.

4.3. Layer Choice

Prior work has shown that acoustic, phonetic, and semantic information is expressed in varying degrees of effectiveness across the layers of a pre-trained speech model (Pasad et al., 2023; Kumar et al., 2022). To assess how layer selection affects Latent-RQ's downstream performance, we investigate different combinations of target layers used as inputs to the random quantizer. Specifically, we extract activations from the stage 0 model at 100k steps and use them to generate quantization targets for continued training over an additional 100k steps. The resulting models are then evaluated on ASR, SID, IC, and ER tasks using the SUPERB benchmark. In addition to the 17 Conformer layers, we include the hidden activation from the initial CNN feature extractor, denoted as layer -1 in Table 2. All configurations use a total of 18 codebooks, distributed equally across the selected layers.

Table 2: Target layers used for quantization (zero-indexed) and corresponding SUPERB results. ASR reports WER (%); SID, ER, and IC report accuracy (%). Best results are in bold; second-best are underlined. Aggregated scores are provided under the Avg column.

Layer Index	ASR	SID	ER	IC	Avg
-1 to 16	**11.86**	68.29	59.88	91.35	76.92
-1 to 4	11.88	67.05	59.75	90.75	76.42
5 to 10	11.97	**68.44**	60.17	91.43	**77.02**
11 to 16	11.97	65.76	60.32	**93.83**	76.99
8	12.24	66.62	60.17	90.25	76.20
2,3,8,9,14,15	12.14	67.33	**60.36**	91.41	76.74

The results clearly indicate that no single set of layers emerges as universally optimal, pointing instead to a strong pattern of task-specific specialization. This is most evident in the trade-off between speaker and intent recognition. The final layers (11-16) achieve the highest IC accuracy at 93.83%, a substantial 2.4% lead over the next best configuration. However, this same configuration yields the lowest score on SID. Conversely, the middle layers (5-10) are best suited for SID, reaching a peak accuracy of 68.44%.

While ASR performance appears to favor a broad range of shallow-to-middle layers, ER shows a slight preference for a sparse, hybrid combination of layers. It is noteworthy that for both ER and ASR, the performance across all configurations is tightly clustered, with total variations of only 0.61% and 0.38 WER respectively, making it difficult to declare a practical winner. Among the evaluated configurations, the middle-layer set (5-10) stands out as the best general-purpose choice. It secures the top score in SID and the second-highest in IC while remaining highly competitive in ASR and ER, providing strong support for our choice (Section 4.1) to use middle layers for general pre-training.

Interestingly, using all layers (-1 to 16) slightly underperforms several more targeted configurations. A plausible explanation is that forcing the model to learn from all layers simultaneously increases task difficulty, particularly given a fixed number of codebooks. Our setup allocates only one codebook per layer in this case, potentially restricting the modeling capacity compared to configurations that assign multiple codebooks to a more focused set of layers. Despite these nuances, it is crucial to note that all multi-layer configurations outperform the single-layer baseline (8), highlighting the value of combining diverse representation levels to guide learning.

4.4. Layer Contribution Dynamics

Building on the layer-wise analysis from the previous section, we further examine how the contributions of individual encoder layers evolve in the downstream models. In particular, we leverage the weighted sum of layers from the SUPERB benchmark to visualize how different pre-training target layer configurations influence the utilization of speech encoder layers during fine-tuning. Fig. 4 presents the normalized contribution weights across layers for ASR, SID, IC, and ER tasks, comparing several layer configurations used to generate latent quantization targets during pre-training.

These visualizations reveal that the choice of target layers affects how information is encoded and distributed across the network, with especially clear effects for SID and IC, and to a lesser extent ASR. For instance, using targets from the middle layers (5–10) or from layer 8 leads to a noticeable shift of IC-related information toward deeper encoder layers, in contrast to models trained directly with Mel spectrogram targets. In the SID task, latent targets sharply concentrate the downstream model's reliance on the final encoder layer (16), whereas Mel-based models draw more evenly from the last few layers. ASR also exhibits an increased emphasis on the final layer when pre-trained with layers 5–10 or layer 8, compared to baseline targets. In general, the use of shallower layers (-1 to 4) produces a distribution more closely resembling that of the Mel spectrogram.

However, a closer look reveals a mismatch between the layers most utilized during downstream fine-tuning and those that yield the best pre-training targets (Table 2). For instance, ASR models rely on layers 8–12, yet earlier layers produce better pre-training performance. SID draws almost entirely from the final layer, despite stronger results from middle-layer targets. Similarly, IC benefits most from layers 11–16, but downstream usage still favors the middle. Only ER shows alignment, consistent with its need for distributed representations. These findings indicate that downstream layer weighted sum profiles do not reliably signal which layers are optimal for target generation during pre-training. The relationship between target design and fine-tuning dynamics remains complex, warranting further investigation into how information is encoded and transformed across training phases.

Finally, we compare these observations to similar visualizations for HuBERT and WavLM from Chen et al. (2021). For ASR and IC, the contribution distributions in our Latent-RQ models closely resemble those of HuBERT Base, while WavLM Base+ tends to shift more

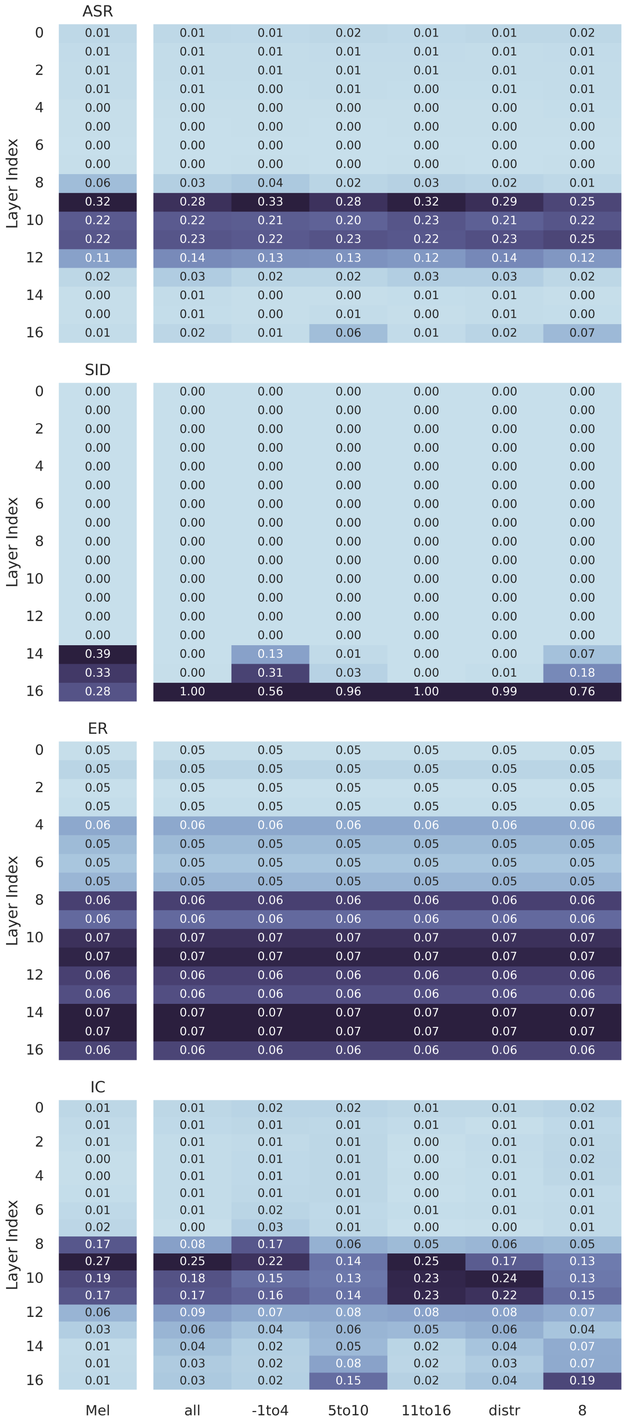


Figure 4: Heatmaps representing the contribution of each Conformer layer towards the input of the downstream model. Darker colors indicate bigger contributions, although each task is separately scaled. The baseline using Mel spectrogram-derived targets is on the left. Other configurations correspond to those in Table 2, with *all* referring to layer set -1 to 16 and *distr* to layers 2,3,8,9,14,15.

weight toward deeper layers. The SID task reveals the biggest contrast; HuBERT Base and WavLM Base+ rely more heavily on early and middle layers, whereas Latent-RQ concentrates almost exclusively on the final encoder layer. This pattern is more akin to what is observed in the Large model variants, where the last layer dominates, although early-layer contributions remain non-trivial in their case. As ER was not analyzed in Chen et al. (2021), our work is, to our knowledge, the first to provide insight into layer-wise information distribution for this task.

4.5. Target Representation Quality

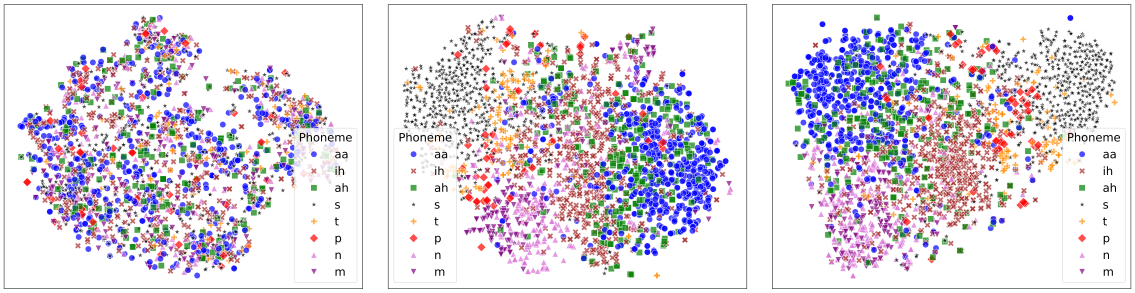


Figure 5: t-SNE plots of phonetic features from the TIMIT dataset derived using random projections from Mel spectrograms (left), stage 1 latent features at 200k steps (middle) and stage 2 at 500k steps (right). Different phonetic labels are demarcated by color and shape.

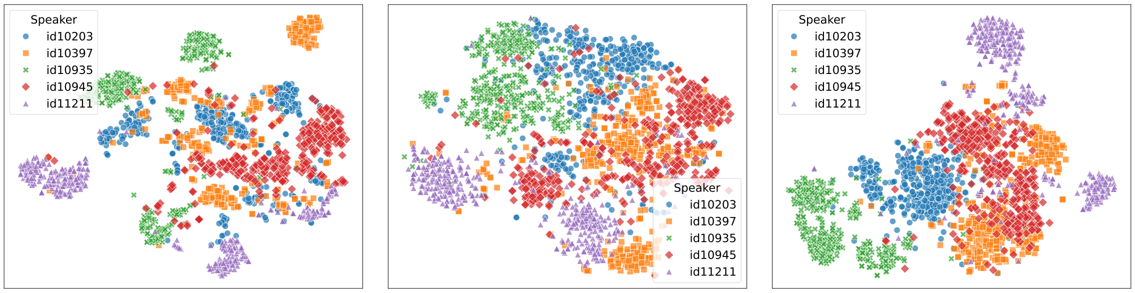


Figure 6: t-SNE plots of speaker features from the VoxCeleb dataset derived using random projections from Mel spectrograms (left), stage 1 latent features at 200k steps (middle) and stage 2 at 500k steps (right). Different speaker labels are demarcated by color and shape.

To assess the quality of target features at different Latent-RQ training stages and in comparison to BEST-RQ, we present t-SNE (Van der Maaten and Hinton, 2008) visualizations of phoneme and speaker embeddings extracted after random projection, that may help explain the improvements seen in the ASR and SID tasks previously. Phoneme-level analysis was performed on TIMIT (Garofolo et al., 1993) using eight diverse phonemes, including vowels (aa, ih, ah), nasals (m, n), stops (t, p), and fricatives (s) from the DR1 test subset (Fig. 5). While Mel-based projections remain highly overlapped since they are fixed during pre-training, encoder-derived features form clearer separation of clusters as early as the 200k-step checkpoint in stage 1, particularly for vowels. Comparing stage 1 and 2, we observe further improvement of phonetic separation as training progresses. By stage 2, clusters for both vowels and more confusable sounds like nasals and stops become better separated, reflecting enhanced phonetic discrimination.

Speaker-level t-SNE plots (Fig. 6) were derived from temporal mean-pooling of the random projections using 400 utterances each from five randomly selected speakers of the VoxCeleb1 development set (Nagrani et al., 2017). Similar to phonemes, they show increasingly aligned speaker clusters going from Mel to latent targets. Stage 2 embeddings yield the most compact and coherent speaker clusters, indicating better disentanglement of speaker identity in Latent-RQ targets. These results support our hypothesis that using latent representations as quantizer inputs enhances the discriminability of learned features, thereby improving targets over the pre-training period and against the BEST-RQ baseline.

5. Conclusion

We introduced Latent-RQ, a scalable extension of BEST-RQ that replaces Mel spectrograms with intermediate latent representations as inputs to a random quantizer. This modification produces more effective SSL targets, reducing both the number of optimization steps and the overall training time needed to match the performance of the baseline BEST-RQ model. Across four downstream tasks, Latent-RQ achieved superior results in three, most notably in ASR and SID. By periodically updating the target encoder during training, Latent-RQ refines its targets in tandem with model improvement, providing a lightweight alternative to the iterative re-labeling procedures employed in HuBERT and WavLM.

Our analyses show that downstream performance is sensitive to both the choice of encoder layers and the timing of target updates, with no single configuration proving optimal across all tasks. Weighted sum analysis and t-SNE visualizations further illustrate how learned targets reshape information flow within the encoder and enhance target quality. While this work demonstrates that stronger speech SSL models can be trained more efficiently with fewer optimization steps and reduced compute, future directions include exploring whether Latent-RQ can also reduce data requirements for pre-training and whether its advantages persist at larger training scales. Additional research may also investigate the impact of target encoder update frequency and scheduling strategies, as well as adaptive layer selection methods beyond fixed configurations.

Acknowledgments

This research is supported by the National Research Foundation, Singapore under its National Large Language Models Funding Initiative. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not reflect the views of the National Research Foundation, Singapore.

References

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

Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

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

Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(8):1798–1828, 2013.

Li-Wei Chen, Takuya Higuchi, He Bai, Ahmed Hussen Abdelaziz, Shinji Watanabe, Alexander Rudnicky, Tatiana Likhomanenko, Barry-John Theobald, and Zakaria Aldeneh. Exploring prediction targets in masked pre-training for speech foundation models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Micheal Zeng, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16:1505–1518, 2021. URL https://api.semanticscholar.org/CorpusID:239885872.

William Chen, Xuankai Chang, Yifan Peng, Zhaoheng Ni, Soumi Maiti, and Shinji Watanabe. Reducing barriers to self-supervised learning: HuBERT pre-training with academic compute. In *Interspeech*, pages 4404–4408, 2023. doi: 10.21437/Interspeech.2023-1176.

William Chen, Wangyou Zhang, Yifan Peng, Xinjian Li, Jinchuan Tian, Jiatong Shi, Xuankai Chang, Soumi Maiti, Karen Livescu, and Shinji Watanabe. Towards robust speech representation learning for thousands of languages. In *Conference on Empirical Methods in Natural Language Processing*, pages 10205–10224, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.570. URL https://aclanthology.org/2024.emnlp-main.570/.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In *International Conference on Machine Learning*, pages 3915–3924. PMLR, 2022.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. w2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250. IEEE, 2021.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, 2019.

John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathan G. Fiscus, David S. Pallett, and Nancy L. Dahlgren. TIMIT acoustic-phonetic continuous speech corpus. https://catalog.ldc.upenn.edu/LDC93S1, 1993. Linguistic Data Consortium, Philadelphia.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*, pages 5036–5040. ISCA, 2020. URL http://dblp.uni-trier.de/db/conf/interspeech/interspeech2020.html#GulatiQCPZYHWZW20.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. HuBERT: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460, 2021. ISSN 2329-9290. doi: 10.1109/TASLP.2021.3122291. URL https://doi.org/10.1109/TASLP.2021.3122291.

He Huang, Taejin Park, Kunal Dhawan, Ivan Medennikov, Krishna C Puvvada, Nithin Rao Koluguri, Weiqing Wang, Jagadeesh Balam, and Boris Ginsburg. NEST: Self-supervised fast conformer as all-purpose seasoning to speech processing tasks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2025.

Muhammad Huzaifah, Tianchi Liu, Hardik B Sailor, Kye Min Tan, Tarun K Vangani, Qiongqiong Wang, Jeremy H M Wong, Nancy F Chen, and Ai Ti Aw. MERaLiON-SpeechEncoder: Towards a speech foundation model for singapore and beyond. *arXiv preprint arXiv:2412.11538*, 2024.

William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26(189-206):1, 1984.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. SpanBERT: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

Pratik Kumar, Vrunda N Sukhadia, and Srinivasan Umesh. Investigation of robustness of HuBERT features from different layers to domain, accent and language variations. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6887–6891. IEEE, 2022.

Abdelrahman Mohamed, Hung-yi Lee, Lasse Borgholt, Jakob D. Havtorn, Joakim Edin, Christian Igel, Katrin Kirchhoff, Shang-Wen Li, Karen Livescu, Lars Maaløe, Tara N. Sainath, and Shinji Watanabe. Self-supervised speech representation learning: A review. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1179–1210, 2022. doi: 10.1109/JSTSP.2022.3207050.

Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. VoxCeleb: A large-scale speaker identification dataset. In *Interspeech*, pages 2616–2620, 2017.

Tu Anh Nguyen, Benoit Sagot, and Emmanuel Dupoux. Are discrete units necessary for spoken language modeling? *IEEE Journal of Selected Topics in Signal Processing*, 16(6): 1415–1423, 2022.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An ASR corpus based on public domain audio books. In *IEEE International Conference on Acoustics, Speech and Signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

Ankita Pasad, Bowen Shi, and Karen Livescu. Comparative layer-wise analysis of self-supervised speech models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023.

Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

Ryan Whetten, Titouan Parcollet, Marco Dinarelli, and Yannick Estève. Open implementation and study of BEST-RQ for speech processing. *ICASSP Workshop on Self-supervision in Audio, Speech and Beyond*, pages 460–464, 2024.

Qiantong Xu, Tatiana Likhomanenko, Jacob Kahn, Awni Y. Hannun, Gabriel Synnaeve, and Ronan Collobert. Iterative pseudo-labeling for speech recognition. In *Interspeech*, pages 1006–1010, 2020. URL https://api.semanticscholar.org/CorpusID:218684543.

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. SUPERB: Speech Processing Universal PERformance Benchmark. In *Interspeech*, pages 1194–1198, 2021. doi: 10.21437/Interspeech.2021-1775.

Yu Zhang, Wei Han, James Qin, Yongqiang Wang, Ankur Bapna, Zhehuai Chen, Nanxin Chen, Bo Li, Vera Axelrod, Gary Wang, Zhong Meng, Ke Hu, Andrew Rosenberg, Rohit Prabhavalkar, Daniel S. Park, Parisa Haghani, Jason Riesa, Ginger Perng, Hagen Soltau, Trevor Strohman, Bhuvana Ramabhadran, Tara Sainath, Pedro Moreno, Chung-Cheng Chiu, Johan Schalkwyk, Françoise Beaufays, and Yonghui Wu. Google USM: Scaling automatic speech recognition beyond 100 languages. *arXiv preprint arXiv:2303.01037*, 2023.