Explainable Disentanglement on Discrete Speech Representations for Noise-Robust ASR

Shreyas Gopal, Ashutosh Anshul, Haoyang Li, Yue Heng Yeo, Hexin Liu, Eng Siong Chng
College of Computing and Data Science, Nanyang Technological University, Singapore
E-mail: shreyas011@e.ntu.edu.sg

***Abstract*—Discrete audio representations are gaining traction in speech modeling due to their interpretability and compatibility with large language models, but are not always optimized for noisy or real-world environments. Building on existing works that quantize Whisper embeddings for speech-to-unit modeling, we propose disentangling semantic speech content from background noise in the latent space. Our end-to-end model separates clean speech in the form of codebook tokens, while extracting interpretable noise vectors as quantization residue which are supervised via a lightweight classifier. We show that our approach improves alignment between clean/noisy speech and text, producing speech tokens that display a high degree of noise-invariance, and improves ASR performance. Keeping Whisper frozen, we show an 82% reduction in error rate compared to Whisper, and 35% improvement over baseline methods on the VBDemand test set. Further analyses show that the learned token space generalizes well to both seen and unseen acoustic conditions.**

I. Introduction

Recent advances in speech modeling have increasingly adopted discrete audio representations for tasks such as automatic speech recognition (ASR), text-to-speech (TTS), and speech-language modeling. Existing works on neural audio codecs, such as DAC [1], EnCodec [2], and SoundStream [3], focus on compressing audio into discrete tokens for high-quality reconstruction and low-bitrate transmission. Works like [4], [5] demonstrate that with task-supervised training, these discrete representations can also yield competitive ASR results. Similarly, methods like [6]–[8] have shown that vector-quantized (VQ) tokens can capture compressed and interpretable speech units for TTS and speech generation. This trend is also motivated by the natural compatibility between discrete audio and tokenized sequences used in large language models (LLMs). For example, HuBERT-based speech representations [9] combined with LLMs can achieve strong ASR performance, as shown in [6], [10]. Unfortunately, a key issue with these VQ-based approaches is the information loss introduced by quantization, which can hinder downstream performance, particularly under noisy or adverse conditions.

Most prior works focus on representing clean speech, which does not reflect real-world noisy scenarios, highlighting a gap in noise-robust discrete representations. Although OpenAI's Whisper [11] exhibits strong performance under clean or moderately noisy conditions, [12], [13] show that its continuous latent representations still encode background noise, which may affect downstream tasks. This issue is echoed in Speechless [14], which highlights a performance gap between clean and noisy inputs when using a similar VQ-module on Whisper embeddings. These observations suggest that quantized representations align well with clean inputs but fail to generalize well under noisy or real world settings, leading to semantic degradation. We posit that ASR supervision and semantic alignment alone are insufficient to address this issue. Instead, jointly supervising both semantic and noise representations provides a more effective path to robust speech modeling.

Aligned with this hypothesis, we introduce a vector quantization (VQ) module that separates clean speech features from noise in Whisper's latent space. We freeze the Whisper encoder and decoder and train lightweight modules that align with clean targets while guiding the quantization residue to capture noise. Our key idea is to frame disentanglement as a vector difference: the quantization residue is treated as an explicit and interpretable representation of background noise.

A. Related Works

Methods such as [15], [16] aim to directly improve speech encoder robustness to noise using contrastive losses. Several other approaches attempt to disentangle semantics from noise. Omran et al. [17] and Bie et al. [18] allocate partitions or separate codebooks to semantic and non-semantic components using architectural or label-based supervision. De'hubert [19] applies cross-correlation and contrastive losses on HuBERT embeddings to align semantically similar sentences injected with different noise conditions.

In contrast to approaches that rely on partitioning or multiple codebooks, we use a single codebook to capture speech semantics. Although traditional RVQ methods quantize the residue at multiple stages, our setup uses a single-stage quantizer and interprets the unquantized residue as an explicit estimate of background noise. A lightweight classifier supervises this residue without mapping it to discrete tokens. While De'hubert relies on fine-tuned HuBERT embeddings and shows gains in ASR performance, our system benefits from Whisper's ASR pretraining, providing a strong semantic prior.

B. Our Contributions

In summary:

- We propose an end-to-end model that disentangles speech from noise in the latent space of Whisper [11] using a vector quantizer that captures semantic features and treats quantization error as noise.
- We define a compound loss that guides both clean alignment and noise extraction in an explainable manner.

979-8-3315-7206-8/25/$31.00

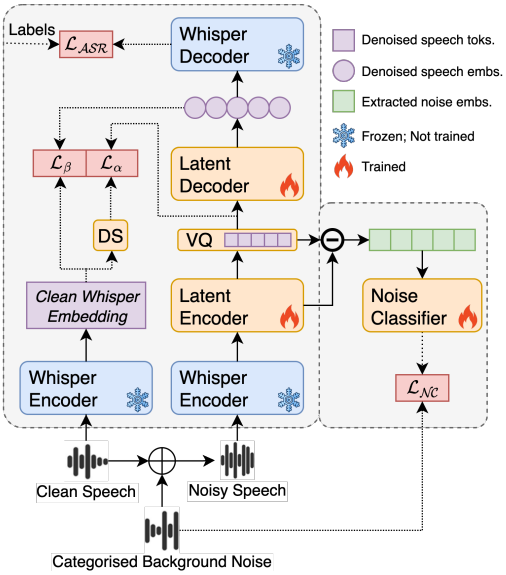

Fig. 1. Proposed model architecture. The dotted rectangles highlight two main components: (1) an extended speech encoder (whisper encoder and latent encoder) and latent decoder that improves alignment between clean speech tokens and text, and (2) a noise disentanglement module that guides the quantization residue to model background noise.

- We show that the quantization residue is interpretable and can be explicitly classified, while also demonstrating robustness to unseen noise.
- Despite information loss due to imperfect disentanglement, our method achieves competitive ASR performance, preserving useful semantic information.

II. Methodology

We build on the work of Speechless [14], a speech-to-unit model using residual vector quantization (RVQ). Our model uses the pretrained *whisper-medium* encoder to provide latent speech embeddings, and the Whisper encoder and decoder remain frozen during training. Fig. 1 shows the overall architecture, and in this section we explain the underlying modules.

A. Extended Encoder

We use paired clean and noisy monophonic signals, $X' \in \mathbb{R}^{1\times T}$ and $X \in \mathbb{R}^{1\times T}$, containing the same semantic content. The clean signal is only used for reference in loss computation, while only the noisy signal is passed through the model.

1) Whisper Encoder: The frozen Whisper encoder outputs $W(X'), W(X) \in \mathbb{R}^{T'\times D}$, where $T' = 1500$ denotes the sequence length and $D = 1024$ the embedding dimension.

2) Latent Encoder: As shown in Fig. 2, the latent encoder consists of a downsampling module, followed by a projection layer $P_d \in \mathbb{R}^{1024\times 64}$. Downsampling reduces the temporal resolution by a factor of 2, such that each token encodes 40ms of audio. In the english language, phonemes range from a duration of 80-120ms with an average around 100ms [20]. This implies that our tokens are sub-phonetic, and sequences of tokens make up phonemes and words. We experiment with mean-pooling, strided 1D convolution, and conv-transformer

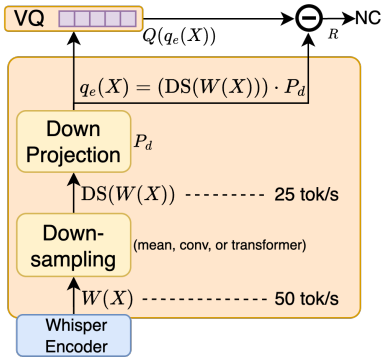

Fig. 2. Latent encoder (orange). Whisper produces 50 embeddings/sec, while our encoder downsamples to 25 tokens/sec.

from [21], with the latter performing best. As trainable parameters in the encoder are important for embedding placement, for pooling and conv1d, we further use an MLP block. The output is seen in (1).

$$q_e(X) = \mathrm{DS}(W(X)) \cdot P_d \tag{1}$$

B. Vector Quantization

VQ creates a discrete bottleneck that captures task-relevant semantic information, yielding quantized embeddings $Q(q_e(X)) \in \mathbb{R}^{750\times 64}$, derived from codebook $\mathcal{C} \in \mathbb{R}^{N_{\text{CB}}\times 64}$ of size N_{CB}. Here, $q_e(X)$ denotes the output of the latent encoder. $Q(\cdot)$ performs the nearest-neighbor lookup in $\mathcal{C}$, whose codes represent denoised speech embeddings, which are further used for noise disentanglement and in the latent decoder. As a straight-through estimator is used, gradients will not directly impact codebook embeddings. Instead estimated moving averages are used to update the codebook with a codebook decay of 0.9. The optimal value of N_{CB} varies with the size of the training corpus and its associated vocabulary. In our case $N_{\text{CB}} = 1024$ showed the best results.

Sequence information is preserved by 1) using fixed codebook token as padding token and 2) restoring positional encoding; This is not required when down-sampling using a conv-transformer as we find it is able to retain sequence information and embed all padded timesteps close together.

1) Noise Disentanglement: Prior work shows that latent vector arithmetic can encode semantic differences—e.g., CSVAE [22] and attribute-aligned VAEs [23]. We leverage this by treating the quantization residue R as a representation of noise, and $Q(q_e(X))$ as capturing semantic content. This disentanglement is seen in (2).

$$R = q_e(X) - Q(q_e(X)) \tag{2}$$

We guide this separation using auxiliary loss terms that push $Q(q_e(X))$ toward clean semantics and R toward noise. Inspired by [12], we apply a lightweight classifier (optionally transformer-augmented) to R for noise classification. This supervision enforces extraction of noise features in R.

C. Latent Decoder

The latent decoder $D_L(\cdot)$ reconstructs Whisper-like embeddings from the quantized outputs. It consists of a linear projection $P_u \in \mathbb{R}^{64\times1024}$, an up-sampling module, and a transformer refinement block. P_u restores the embedding dimension, and the up-sampler restores sequence length T'. We compare repeat and transposed 1D convolution for up-sampling. The simpler repeat method, where tokens are duplicated in time, performed better in our setup (see Table I). The transformer refinement block aids in smoothing artifacts due to up-sampling and aligning with Whisper's latent space.

D. Training Cost Function

We employ a loss function that combines four components: ASR cross-entropy ($\mathcal{L}_{\text{ASR}}$), VQ commitment loss ($\mathcal{L}_\alpha$), a semantic disentanglement loss ($\mathcal{L}_\beta$), and an auxiliary noise classification loss ($\mathcal{L}_{\text{NC}}$). The total cost is defined as:

$$\mathcal{L} = \mathcal{W}_{\text{ASR}} \cdot \mathcal{L}_{\text{ASR}} + \mathcal{W}_\alpha \cdot \mathcal{L}_\alpha + \mathcal{W}_\beta \cdot \mathcal{L}_\beta + \mathcal{W}_{\text{NC}} \cdot \mathcal{L}_{\text{NC}} \quad (3)$$

Through experiments, we found that $\mathcal{W}_\alpha + \mathcal{W}_\beta = 1.0$ gave the most consistent results, and setting $\mathcal{W}_{\text{ASR}} = 1.0$ provides a stable training direction early-on when quantizer outputs are still noisy. Table II shows the relevance of each loss term.

1) ASR Cross-Entropy Loss: Given a noisy speech signal X, which passes through the encoder, quantizer, and decoder to yield the model output $q_d(Q(q_e(X)))$, and a sequence of previous tokens $y_{t-1}, ..., y_1$, the cross-entropy loss aims to minimize the error in predicting the next token y_t:

$$\mathcal{L}_{\text{ASR}} = -\sum_{t=1}^{T} \log \mathcal{P}_\theta(y_t | y_{t-1}, ..., y_1, q_d(Q(q_e(X)))) \quad (4)$$

This term is particularly important during the initial phases of training when quantizer embeddings are still adapting.

2) Commitment Loss: As introduced in foundational VQ-VAE works [24], the commitment loss prevents the uncontrolled growth of encoder outputs and stabilizes training. In our case, we define this loss as the L2 distance between the downsampled clean embeddings, $\text{DS}\left[q_e(X')\right]$ and the quantized outputs from the noisy signal:

$$\mathcal{L}_\alpha = \|\text{DS}\left[q_e(X')\right] - Q(q_e(X))\|_2^2 \quad (5)$$

This encourages the noisy embeddings to stay close to their clean counterparts in latent space. The best performance was observed when $\mathcal{W}_\alpha = 0.5$.

3) Semantic Disentanglement Loss: This term corrects quantization artifacts by guiding the refined output toward the original Whisper latent space. It is defined as the L2 distance between the reference clean embeddings $q_e(X')$ and the output of the latent decoder:

$$\mathcal{L}_\beta = \|q_e(X') - D_L(Q(q_e(X)))\|_2^2 \quad (6)$$

We found that $\mathcal{W}_\beta = 0.5$ worked best when paired with the commitment loss.

4) Noise Classification Loss: To explicitly model the noise component, we compute the quantization residue as seen in (2) and pass it through a lightweight classifier $NC(\cdot)$ to predict the noise class Y'. The loss is defined as:

$$\mathcal{L}_{\text{NC}} = -\sum \log P_{\theta_{\text{NC}}}(Y' | NC(R)) \quad (7)$$

We found that scheduling $\mathcal{W}_{\text{NC}}$ inversely proportional to the classifier's validation accuracy helped prevent overfitting and improved generalization.

III. Experiments

A. Datasets

1) VBDemand: VBDemand [25] is widely used for speech enhancement and robust ASR [26]. It includes clean-noisy speech pairs with transcriptions. We combine two subsets: one with 11,572 utterances (9.5h, English accents) and another with 21,225 utterances (19h, US and Scottish accents). Of this 28.5h, we reserve 1.5h (disjoint speakers) for validation and use the rest for training. Each training and validation sample contains 1 of 10 noise types. The noisy test set has 5 unseen noise types: bus, cafe, living room, office, and pedestrian.

2) CHiME-4: We use the `test-real` and `test-simu` splits from CHiME-4 [27] for additional qualitative analysis to assess the generalization ability of our model under out-of-distribution noise and speaker conditions, as the clean-noisy pairs are perfectly time-aligned with various noise types.

B. Setup

All experiments were conducted on 2 A100 GPUs with a total batch size of 64 (32 per GPU). We used a linear learning rate warm-up for the first 500 steps, followed by a cosine annealing schedule. The initial learning rate was set to $1e^{-3}$, and optimization was performed using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.95$) until convergence. We initialize the whisper components using pretrained *whisper-medium* checkpoints from the official OpenAI Whisper repository[1]. Codebooks were initialized using Kaiming noise as described in [14]. All WER results presented are derived using autoregressive decoding with a beam-size of 10 and temperature of 0.0.

C. Evaluation

1) Metric: As the Whisper decoder's final objective is next-token prediction, we correlate ASR performance with the degree of semantic information retained in the latent embedding. We use word error rate (WER) as the primary evaluation metric for both our model and the baselines, and use relative error reduction (RER %) to compare performances.

2) Baseline: We use the frozen *whisper-medium* model on the VB-Demand test split to establish a performance baseline. Given that 1) Our method inserts adapter-style modules between the Whisper encoder and decoder without fine-tuning either and 2) a degree of semantic loss is inevitable in our approach due to quantization; We consider an additional baseline model (+Adapter in Table I) where we insert

[1]https://github.com/openai/whisper

TABLE I

ASR PERFORMANCE (WER) OF BASELINES, OUR MODEL, AND UPPER BOUND. THE FIRST GROUP VARIES CODEBOOK SIZE WITH FIXED MEAN-REPEAT (MR) SAMPLING. THE SECOND GROUP COMPARES VARIOUS SAMPLING CONFIGURATIONS USING THE BEST CODEBOOK SIZE ($N_{\text{CB}} = 1024$.)

WER ↓	**Baseline**			**Codebook Size (N_{CB}) vs. WER**					**Sampling Strategies vs. WER**						**Upper Bound**	
VBDemand	Whisper	+Adapter	Speechless	128	256	512	**1024**	2048	MR	MC	CC	CR	**TR**	TC	C_{ZS}	C_{TR}
Validation																
all	6.75	**0.35**	-	0.58	0.31	0.29	**0.21**	0.49	0.21	**0.19**	0.24	0.28	*0.64*	0.32	2.47	0.68
Test																
bus	12.05	10.68	-	2.58	**1.86**	1.99	2.67	3.24	2.67	**1.62**	2.91	2.10	1.86	1.86	-	-
cafe	25.71	4.03	-	6.25	5.30	4.97	**4.59**	5.54	4.59	**3.88**	4.83	5.22	**3.88**	4.51	-	-
living	8.31	2.10	-	4.04	3.31	**3.30**	3.47	4.77	3.47	4.03	3.07	3.63	**2.75**	3.72	-	-
office	9.01	1.18	-	2.51	2.04	1.93	**1.57**	2.98	**1.57**	2.12	2.04	2.35	1.65	1.65	-	-
pedestrian	13.44	1.15	-	3.30	2.15	2.14	**1.84**	2.30	1.84	2.30	2.15	**1.61**	2.23	1.46	-	-
all	13.72	**3.78**	12.34	3.74	2.93	2.87	**2.82**	3.75	2.82	2.79	2.99	2.98	**2.47**	2.63	12.79	1.87

Legend: Whisper = Whisper-medium zero-shot, +Adapter = Whisper-medium with trainable linear adapter; MR = (mean, repeat), MC = (mean, conv1d), CC = (conv1d, conv1d), CR = (conv1d, repeat), TR = (conv-transformer, repeat), TC = (conv-transformer, conv1d); C_{ZS} = Whisper-medium zero-shot with Clean speech input C_{TR} = Our model (1024-codebook TR setting) with Clean speech input

TABLE II

ABLATION STUDY OF LOSS TERMS. ALL CONFIGURATIONS USE WHISPER-MEDIUM, $N_{\text{CB}} = 512$, AND (MEAN, REPEAT) DS/US SETUP.

Model	**Losses Used**	**DS/US**	N_{CB}	**WER (%) ↓**
Ours	Only VQ and $\mathcal{L}_{\text{ASR}}$	MR	512	3.89
	+ $\mathcal{L}_{\text{NC}}$	MR	512	3.63
	+ Only $\mathcal{L}_{\alpha}$	MR	512	3.36
	+ Only $\mathcal{L}_{\beta}$	MR	512	3.11
	+ $\mathcal{L}_{\alpha}$ + $\mathcal{L}_{\beta}$	MR	512	**2.87**

trainable adapters—without discretization—between encoder and decoder. Each adapter block consists of 2 MLP layers with a GELU activation function between them. This black-box adapter provides a non-interpretable baseline approach that does not suffer info loss due to quantization. Lastly, the authors of Speechless [14] show that VQ-VAE training with supervision only for semantic alignment results in poor performance in noisy conditions, hence we include results from their 2560-codebook model under similar conditions.

3) Upper Bound: We consider the performance with clean speech inputs as the upper-bound. The VBDemand clean test-set is used as the input and we tabulate zero-shot inference using Whisper-medium (C_{ZS}) as well as our best model's performance (C_{TR}: $N_{\text{CB}} = 1024$ with conv-tranformer down-sampling and repeat upsampling) with clean inputs.

D. Primary Experiments

We show that three factors contribute to the effectiveness of our method: (1) the VQ codebook size, (2) the number of trainable parameters before VQ, and (3) the choice of down-sampling and upsampling strategies. (2) and (3) are tightly coupled as layers preceding the VQ module are expected to improve embedding placement and help the model learn meaningful representations while also performing downsampling.

In our first study, we fixed the downsampling to simple mean pooling and upsampling to repeat (MR), varying only the codebook size. After identifying $N_{\text{CB}} = 1024$ as optimal, we studied the effect of different sampling strategies on ASR and NC performance. These strategies are shown in Table I.

Mean pooling relies heavily on MLP layers for embedding placement. Conv1D-based downsampling introduces additional learnable filters, while conv-transformer modules operate over both time and feature dimensions, enabling more complex token positioning. These results are presented in Table I. We further discuss these results in detail in Section IV-A.

E. Ablation Study of Loss Terms

To study the contribution of each loss term, we perform an experiment where loss components are added incrementally. This illustrates how each loss term contributes towards improving the model's performance. Results are shown in Table II.

IV. RESULTS AND DISCUSSION

A. Quantitative Observations

We present our findings in Table I. The baseline whisper-medium model achieves a WER of 13.72% on the VBDemand noisy test set. Adding trainable linear adapters further reduces the WER to 3.78% (+Adapter; where no quantization info loss occurs). As shown, the majority of our configurations show performance gains compared to inserted linear MLP adapters, with our best approach (N_{CB}=1024, TR) achieving a WER of 2.47%. This shows a relative error reduction of 82% over the zero-shot whisper baseline, 35% over the trainable adapter and 80% over the base Speechless [14] model. The conv-transformer method enables strong embedding placement requiring 28% fewer epochs for convergence, while also showing the least amount of overfitting to the validation dataset.

Further, in Table II, we analyse the significance of each loss term. The first setting only employs ASR task supervision resulting in a WER of 3.89% which is poorer than adapter training. An RER of 6.7% is achieved by adding only noise disentanglement. The complete loss function results in a WER of 2.87% and total RER of 20.9%, showing that each term of our proposed loss function plays a significant role.

B. Qualitative Observations

1) Noise Interpretation: Fig. 4 (top) shows T-SNE projections of representations from the penultimate layers of the noise-classifier of noisy samples from the validation set after model convergence. We observe that the information removed

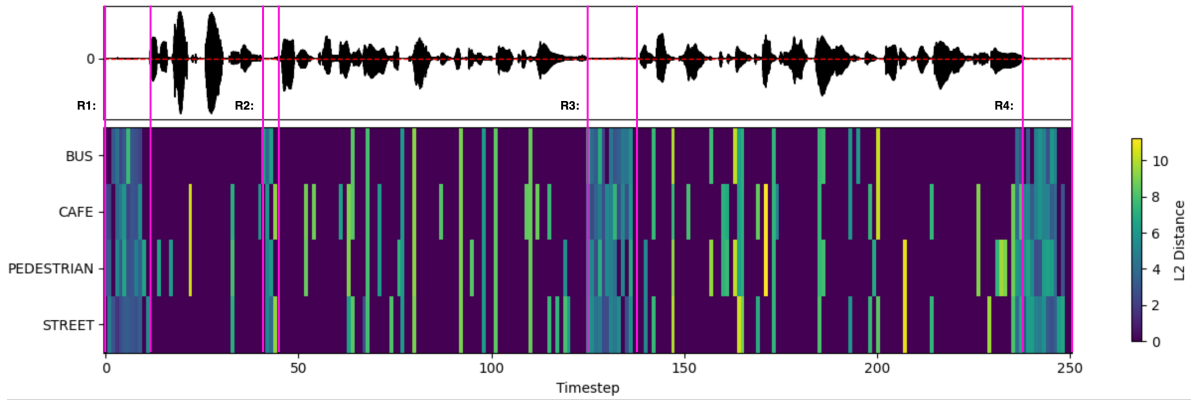

Fig. 3. Mean L2 Distance: Clean vs Noisy Embeddings: (top) Clean audio waveform from Chime-4 test dataset. The red dotted line represents an amplitude of 0. (bottom) The color at each position represents L2 distance of embeddings from our encoder between the clean signal and the clean signal mixed with various background noises. The y-axis indicates the mixed-in background noise, and the x-axis represents the time-steps. Lower values show that the encoder is able to generate more noise invariant representations for sub-word tokens and token sequences similar to those captured from clean speech.

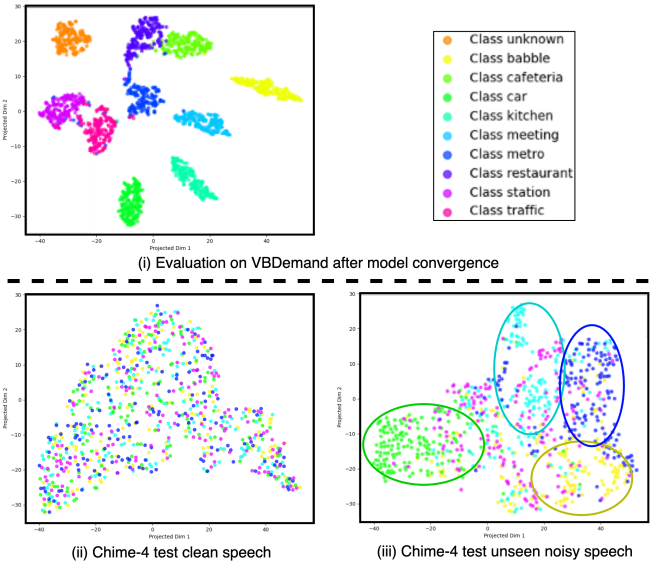

Fig. 4. (Top) T-SNE projections of penultimate-layer features from the noise classifier on VBDemand validation samples. Some samples were relabeled as "unknown" to encourage generalization. These noise types are seen during training. (Bottom) Similar projections from clean and noisy CHiME-4 test samples. Clean speech yields uniform embeddings; noisy speech shows class-specific clustering. CHiME-4 noise types include café (green), bus (light blue), pedestrian (yellow), and street (dark blue), which are unseen during training.

via the quantization error is clearly noise-class separable, and correlates directly with a distribution that helps the classifier categorize the background noise. The final noise classification accuracy for the VBDemand validation dataset is 98.23%.

2) Clean vs. Noisy Speech: Table I shows that our best model (TR) has a much lower WER of 1.87% with clean speech inputs (C_{TR} > TR). This improved performance indicates that our model correctly encodes only semantic data, generalizing well to clean speech. Otherwise performance in clean speech would degrade due to information loss. We also perform inference on clean speech embeddings from the CHiME-4 dataset and as seen in Fig. 4 (bottom-ii), disentangled noise embeddings are not class-separable when the model is input with clean speech. Whereas, as seen in Fig. 4 (bottom-ii), even unseen noisy speech from CHiME-4 yields a good degree of class separability. These observations reinforce that our model separates out only valid noise information when available, while semantic information is correctly tokenized.

3) Noise Invariance: To examine the noise invariance of our latent representations, we examined embeddings of the same speech in the clean setting and with various noise samples mixed. Ideally the embeddings should be similar (invariant) at each time-step. To visualize this we make use of L2 distance of codebook embeddings of the clean and noisy speech. Fig. 3 shows both the clean audio sample from the CHiME-4 test dataset, along with L2 distances between each noisy sample and the original clean sample at each timestep. We see that the majority of embeddings and sequences of embedding show a high degree of noise invariance (dark purple). Patches of high embedding variance correlate directly with silence where the quantizer may not have enough information to generate useful embeddings (R1, R2, R3, R4). Localized errors occur across many noise types and seem to be due to fricative sounds and because CHiME-4 is outside of the training distribution.

V. Conclusions and Future Work

In this work, we propose a novel disentanglement framework to improve the noise-robustness of discrete speech representations, by separating semantic speech tokens from background noise using vector quantization over Whisper embeddings. Without fine-tuning the Whisper encoder, our method achieves an 82% error reduction compared to zero-shot Whisper and a 35% improvement over adapter-based baselines, demonstrating the benefit of dual supervision over semantic and noise representations. Unlike prior methods such as Speechless that align only semantic content, our model jointly learns semantic and noise representations, yielding significantly better performance across noisy conditions (80% reduction in WER). The quantization residue serves as an interpretable noise embedding, enabling accurate classification and an explainable latent space. Visual analysis confirms that the model generalizes well to unseen noise, while preserving clean speech performance.

Future works can leverage the noise residue space for applications such as noise-guided training data augmentation, robust knowledge distillation, and enhancement of pre-trained discrete representation models.

ACKNOWLEDGMENT

The computational work for this article was partially performed on resources of the National Supercomputing Centre, Singapore (https://www.nscc.sg) and partially supported by the High Performance Computing Centre of Nanyang Technological University, Singapore.

REFERENCES

[1] R. Kumar, P. Seetharaman, A. Luebs, I. Kumar, and K. Kumar, "High-fidelity audio compression with improved rvqgan," *Proc. NeurIPS*, vol. 36, pp. 27 980–27 993, 2023.

[2] A. Défossez, J. Copet, G. Synnaeve, and Y. Adi, "High fidelity neural audio compression," *arXiv preprint arXiv:2210.13438*, 2022.

[3] N. Zeghidour, A. Luebs, A. Omran, J. Skoglund, and M. Tagliasacchi, "Soundstream: An end-to-end neural audio codec," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 30, pp. 495–507, 2021.

[4] K. Dhawan, N. R. Koluguri, A. Jukić, R. Langman, J. Balam, and B. Ginsburg, "Codec-asr: Training performant automatic speech recognition systems with discrete speech representations," *arXiv preprint arXiv:2407.03495*, 2024.

[5] K. C. Puvvada, N. R. Koluguri, K. Dhawan, J. Balam, and B. Ginsburg, "Discrete audio representation as an alternative to mel-spectrograms for speaker and speech recognition," in *Proc. ICASSP*, IEEE, 2024, pp. 12 111–12 115.

[6] Y. Hono, K. Mitsuda, T. Zhao, K. Mitsui, T. Wakatsuki, and K. Sawada, "Integrating pre-trained speech and language models for end-to-end speech recognition," *arXiv preprint arXiv:2312.03668*, 2023.

[7] Z. Du, Y. Wang, Q. Chen, *et al.*, "Cosyvoice 2: Scalable streaming speech synthesis with large language models," *arXiv preprint arXiv:2412.10117*, 2024.

[8] A. Défossez, L. Mazaré, M. Orsini, *et al.*, "Moshi: A speech-text foundation model for real-time dialogue," Kyutai, Tech. Rep., 2024.

[9] 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 Trans. Audio, Speech, Lang. Process.*, vol. 29, pp. 3451–3460, 2021.

[10] Y. Xu, S.-X. Zhang, J. Yu, Z. Wu, and D. Yu, "Comparing discrete and continuous space llms for speech recognition," *arXiv preprint arXiv:2409.00800*, 2024.

[11] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proc. ICML*, PMLR, 2023, pp. 28 492–28 518.

[12] Y. Gong, S. Khurana, L. Karlinsky, and J. Glass, "Whisper-at: Noise-robust automatic speech recognizers are also strong general audio event taggers," *arXiv preprint arXiv:2307.03183*, 2023.

[13] X. Zhang, Q. Zhang, H. Liu, *et al.*, "Mamba in speech: Towards an alternative to self-attention," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 33, pp. 1933–1948, 2025.

[14] A. Dao, D. B. Vu, H. H. Ha, *et al.*, "Speechless: Speech instruction training without speech for low resource languages," *arXiv preprint arXiv:2505.17417*, 2025.

[15] Y. Wang, J. Li, H. Wang, Y. Qian, C. Wang, and Y. Wu, "Wav2vec-switch: Contrastive learning from original-noisy speech pairs for robust speech recognition," in *Proc. ICASSP*, IEEE, 2022, pp. 7097–7101.

[16] D. Ng, J. Q. Yip, T. Surana, *et al.*, "I2cr: Improving noise robustness on keyword spotting using inter-intra contrastive regularization," in *Proc. APSIPA ASC*, IEEE, 2022, pp. 605–611.

[17] A. Omran, N. Zeghidour, Z. Borsos, F. de Chaumont Quitry, M. Slaney, and M. Tagliasacchi, "Disentangling speech from surroundings with neural embeddings," in *Proc. ICASSP*, IEEE, 2023, pp. 1–5.

[18] X. Bie, X. Liu, and G. Richard, "Learning source disentanglement in neural audio codec," in *Proc. ICASSP*, IEEE, 2025, pp. 1–5.

[19] D. Ng, R. Zhang, J. Q. Yip, *et al.*, "De'hubert: Disentangling noise in a self-supervised model for robust speech recognition," in *Proc. ICASSP*, IEEE, 2023, pp. 1–5.

[20] G. Ma, P. Hu, J. Kang, S. Huang, and H. Huang, "Leveraging phone mask training for phonetic-reduction-robust e2e uyghur speech recognition," *arXiv preprint arXiv:2204.00819*, 2022.

[21] C.-L. Zhang, J. Wu, and Y. Li, "Actionformer: Localizing moments of actions with transformers," in *Proc. ECCV*, ser. LNCS, vol. 13664, 2022, pp. 492–510.

[22] J. Klys, J. Snell, and R. Zemel, "Learning latent subspaces in variational autoencoders," *Proc. NeurIPS*, vol. 31, 2018.

[23] A. Pati and A. Lerch, "Attribute-based regularization of latent spaces for variational auto-encoders," *Neural Comput. Appl.*, vol. 33, pp. 4429–4444, 2021.

[24] A. Van Den Oord, O. Vinyals, *et al.*, "Neural discrete representation learning," *Proc. NeurIPS*, vol. 30, 2017.

[25] C. V. Botinhao, X. Wang, S. Takaki, and J. Yamagishi, "Investigating rnn-based speech enhancement methods for noise-robust text-to-speech," in *9th ISCA speech synthesis workshop*, 2016, pp. 159–165.

[26] Y. Hu, C. Chen, C.-H. H. Yang, *et al.*, "Large language models are efficient learners of noise-robust speech recognition," in *Proc. ICML*, 2024.

[27] E. Vincent, S. Watanabe, J. Barker, and R. Marxer, "The 4th chime speech separation and recognition challenge," *URL: http://spandh. dcs. shef. ac. uk/chime_challenge/(last accessed on 1 August, 2018)*, 2016.