SPARSE CONTINUOUS REPRESENTATION FOR UNPAIRED SPEECH–TEXT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Learning from unpaired speech and text corpora can substantially improve sample efficiency and accessibility of spoken language systems. However, existing approaches for unpaired speech–text learning face a **capacity–compatibility tradeoff**: Dense, continuous representations preserve rich acoustic information but differ significantly with text in representational structures, resulting in low compatibility; Discrete speech units improve compatibility at the cost of information loss. In this work, we propose **sparse continuous representations (SCRs)** as an alternative that preserves rich information while maintaining compatibility. To validate our approach, we consider the task of unsupervised speech recognition (UASR) and synthesis (UTTS) as testbeds, both canonical examples of unpaired speech-text learning. Across seven languages for UASR, our SCR-based systems, SparPUSM and SparCipher, consistently outperforms the previous state-of-the-art systems, yielding an average character error rate reduction of 8-17%, with up to 17% on LibriSpeech. For UTTS, SCR-based approach also outperforms previous approaches, demonstrating the benefit beyond UASR. Code will be released upon acceptance.

1 INTRODUCTION

In real-world multimodal settings, data are often unpaired because pairing typically requires costly human annotations. In speech domain, this is especially common since paired datasets rely on manual transcripts or translations, which are expensive for most languages. Over-reliance on paired resources not only limits inclusion across languages, but also leaves vast unpaired audio and text unused. To move beyond reliance on paired resources, unsupervised speech recognition (UASR) (Glass, 2012; Baevski et al., 2021; Liu et al., 2023a; Tseng et al., 2024; Wang et al., 2026) and unsupervised text-to-speech synthesis (UTTS) (Ni et al., 2022; Liu et al., 2022) tries to learn directly from unpaired speech and text. However, learning from unpaired data is ill-posed without shared structure across modalities (Wang et al., 2023b). Therefore existing UASR approaches rely either on powerful dense continuous features, such as wav2vec 2.0 (Baevski et al., 2020), HuBERT (Hsu et al., 2021) and XEUS (Chen et al., 2024), or discrete speech units Liu et al. (2018); Wang et al. (2023c; 2024).

However, each form of such speech representations has its own drawbacks. Dense continuous features preserve rich linguistic and acoustic details thanks to their high capacity, but they often exhibit mismatch in representational structure with text. This incompatibiity reflect differences in representational geometry, for example, semantics concentrating in a subspace (Liu et al., 2023b; Han & Tang, 2026) or compositional structure that texts do not share (Choi et al., 2026). As a result, UASRs that rely on dense continuous features are sensitive to the modality gap between speech and text, often leading to unstable training Ni et al. (2022); Wang et al. (2023c). Such a gap may stem from phonological-orthographic mismatch, especially without grapheme-to-phoneme (G2P) convertion, and from temporal misalignment introduced by differences in speech and text segmentation. On the other hand, discretization improves compatibility by allowing a shared sequence modeling objective with text like masked token prediction (Wang et al., 2026). But this compatibility comes at the cost of capacity, as discretization can discard fine-grained acoustic or linguistic detail (Défossez et al., 2023).

To better navigate this **capacity–compatibility tradeoff**, we propose **sparse continuous representations (SCRs)** for speech. SCRs are features where only a few meaningful dimensions are active,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

and those active ones can vary *continuously*. Similar sparse representations have been studied in fields such as signal processing (Mallat & Zhang, 1993; Chen et al., 1998), neuroscience (Olshausen & Field, 1996) and mechanistic interpretability (Cunningham et al., 2024) in machine learning, but using them as an *interface for unpaired speech–text learning* is underexplored. The advantage of SCRs for this aim is twofold. First, SCRs had higher capacity than their discrete counterparts because it allows combining a few atoms with continuous weights, rather than relying on a single atom. As a result, SCRs can preserve finer-grained semantic and acoustic details. Second, their sparse support reorganizes dense representations over a small set of atoms, yielding a more structured space the better aligns with text than dense continuous representations. Such a sparse structure can also potentially reduce parameter interference during optimization for UASR, as theoretical analysis has demonstrated for discrete units (Wang et al., 2023b). Building on this perspective, we apply SCRs to two major unpaired speech-text scenarios, UASR and UTTS, replacing the usual discrete or dense representations accordingly. Our contribution is threefold:

- For G2P-free UASR, we propose two SCR-based approaches, **SparPUSM** and **SparCipher**, show that SCRs consistently outperform both discrete and dense approaches across seven languages;
- For G2P-free UTTS, we designed a **speech sparsifier**, which replaces the standard neural speech codec as the speech representation model, and demonstrate that it is more robust to transcript noise and better preserves semantics, leading to lower word error rate (WER) for the synthesized speech;
- We extend the SCR-based framework to the semi-supervised setup with limited paired data and unpaired text from new domains, and show that SCR continues to be beneficial for those settings.

The rest of the paper is organized as follows: Section 2 discusses related works, Section 3 describes the method, Section 4 presents experimental setup and results, and Section 5 concludes.

2 Related Work

Unpaired speech-text learning. Early work on unpaired speech-text learning (Liu et al., 2018; Chen et al., 2019; Baevski et al., 2021; Liu et al., 2023a; Tseng et al., 2024) often relies on generative adversarial nets (GAN) (Goodfellow et al., 2014) and strong self-supervised speech representations (Baevski et al., 2020; Hsu et al., 2021; Chen et al., 2022; 2024). However, such approaches can be unstable due to the sensitivity of GAN to hyperparameters and the incompatibility of dense features with text Wang et al. (2023c). To improve stability, later work replaced the adversarial objectives with explicit distribution matching (Wang et al., 2024) or unimodal language modeling objective on a shared crossmodal architecture (Ni et al., 2025; Wang et al., 2026). These later approaches rely on representations that are more compatible across modalities, often via *discrete units*. While most early work focus on UASR, later works expanded to translation (Wang et al., 2023a), synthesis (Liu et al., 2022), spoken language understanding (Shi et al., 2023), and unpaired modality beyond text (Wang et al., 2023c).

Sparse representation learning. Sparse representation has a long history in signal processing (Mallat & Zhang, 1993; Chen et al., 1998; Aharon et al., 2006; Gregor & LeCun, 2010), motivated by the ideas that high-dimensional signals can often be compactly expressed with only a few active atoms in an overcomplete dictionary. With the emergence of deep learning, subsequent work has explored *sparse autoencoders (SAE)*, which are neural networks capable of learning more expressive sparse representations (Ranzato et al., 2006; Makhzani & Frey, 2014; 2015; Sprechmann et al., 2015; Sreter & Giryes, 2018; Rolfe & LeCun, 2013) and disentangled representations (Tonolini et al., 2020; Lachapelle et al., 2022). Beyond a single modality, others have applied SAE to capture correspondence across paired modalities (Mairal et al., 2012; Gu et al., 2026; Kaushik et al., 2026). More recently, sparse representation has been applied to the hidden representations of large language models, either to improve interpretability (Yun et al., 2021; Bricken et al., 2023; Cunningham et al., 2024; Dunefsky et al., 2024; Rajamanoharan et al., 2024; Gao et al., 2025; Karvonen et al., 2025; Aparin et al., 2026), or to improve model performance on few-shot settings (Mitra et al., 2025; Cho & Hockenmaier, 2025; Mehta et al., 2026).

3 METHOD

3.1 UNPAIRED SPEECH–TEXT LEARNING

In the task of unpaired speech–text learning, the training data consist of two independently collected corpora, a speech corpus $\mathcal{X}$ and a text corpus $\mathcal{Y}$, without instance-level pairing. The general goal of speech–text learning is then to learn a model that relates speech and text from $\mathcal{X}$ and $\mathcal{Y}$ without paired data. For example, in UASR, the goal is to learn a recognizer $y^* : \mathcal{X} \mapsto \mathcal{Y}$ that maps a speech utterance X to a text sequence Y. In UTTS, we aim instead to learn a conditional generative model that produces a speech waveform X given text Y and conditioning S such as speaker or style.

Conceptually, UASR can be decomposed into two coupled sub-problems: First, learn a speech representation that are both capable and compatible with text, or **compatible representation learning**; And second, learning the cross-modal mapping or **decipherment**.

3.2 COMPATIBLE REPRESENTATION LEARNING VIA SPARSE CONTINUOUS REPRESENTATION

We define two representations U and V are *compatible* if there exists a mapping $f \in \mathcal{F}$ for some low-complexity function class $\mathcal{F}$, for example, a low-rank affine mapping, that maps between them with small error. For examples, previous works in UASR based on discrete speech units uses $U = q(X)$ for some quantizer q to map raw speech waveform to discrete units at the same temporal resolution as the text Y, and then U can be approximately mapped to Y via a permutation f, which is then learned during the decipherment stage as $y = q \circ f$ where $\circ$ denotes function composition. However, discrete quantization can be sub-optimal to balance capacity-compatible tradeoff. For an acoustic word of length L with C possible values per position, a flat, lossless discrete coding would require up to C^L clusters, which is impractical. Compositional quantizers like residual or grouped quantizer (Baevski et al., 2020; Yang et al., 2023; Zeghidour et al., 2022; Défossez et al., 2023) can ease this combinatorial growth, but they still discretize the underlying continuous speech signal. Conversely, dense continuous features, while high capacity, are not necessarily compatible with text in our sense either, since there may be no low-complexity map aligning them, as they tend to store semantic information in a low-dimensional subspace Han & Tang (2026) rather than spanning the whole space like one-hot symbolic vectors.

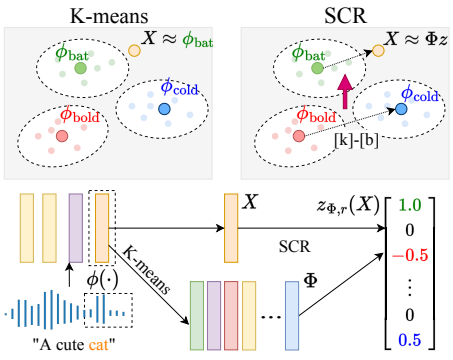


Figure 1: **An intuitive illustration of the potential benefit of sparse continuous representations.** Discrete clustering assign each unit to one bucket, while SCR express new units ("cat") by recombining shared subword components. This is an intuition rather than assumption of our method.

Rather than choosing between quantized units and dense continuous features, we consider the following alternative. First, we start with a dense speech feature sequence $\mathbf{u}(X) = [\mathbf{u}_1(X), \ldots, \mathbf{u}_{m(X)}(X)] \in \mathbb{R}^{d \times m(X)}$ for some variable length $m(X)$ depending on the input utterance, we seek a *sparse encoder* $\mathbf{z}(X) = [\mathbf{z}_1(X), \ldots, \mathbf{z}_{m(X)}(X)] \in \mathbb{R}^{K \times m(X)}$ and an associated dictionary $\mathbf{\Phi} = [\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_K] \in \mathbb{R}^{d \times K}$ such that

$$\mathbf{u}_i(X) = \mathbf{\Phi}\mathbf{z}_i(X) = \sum_{k=1}^{K} z_{ik}(X)\boldsymbol{\phi}_k, \qquad i = 1, \cdots, m(X), \tag{1}$$

where the support of $\mathbf{z}(X)$ selects a discrete subset of *dictionary atoms*, while the nonzero coefficients remain continuous. The representation is sparse in the sense that its support $\|\mathbf{z}_i(X)\|_0 =: r \ll K$ for any $i = 1, \cdots, m(X)$. We call $\mathbf{z}(X)$ the SCR of X. Note that Eq. 1 includes K-means vector quantization as a constrained special case in which $r = 1$ and the single nonzero coefficient is fixed at one. Unlike discrete units, SCRs are able to achieve **compositionality without relying on discrete choices**. In the previous example, a suitable SCR $\mathbf{z}_i(X)$'s can represent all L^C acoustic

word clusters using a much smaller dictionary compared to the flat discrete speech representation. In particular, the dictionary Φ consists of one atom for each position-symbol pair, a total $K = \mathcal{O}(LC)$ atoms. Further, $\mathbf{z}(X)$ only requires at most $r = \mathcal{O}(L)$ active atoms per acoustic word cluster, since it will require at most L "edits" using the atoms, where each edit involves a subtraction between two atoms. A concrete example is illustrated in Figure 1. Therefore, in this case, $\mathbf{z}_i(X)$'s are lower-dimensional than features from a flat quantizer with similar capacity and thus achieve a better capacity-compatibility tradeoff. It also provides a precise mechanism by which sparse composition could exploit the compositionality of language to improve coverage of rare syllables and words, which is consistent with recent works on phonological probing (Choi et al., 2026) on self-supervised speech features.

3.3 Syllable-level speech sparsifier for UASR

To learn Eq. 1 for a given utterance X, we develop a syllable-level speech sparsifier shown in Figure 2. The sparsifier first extract frame-level features using a pretrained self-supervised speech encoder $\mathbf{h}(X)$:

$$\mathbf{h}(X) := [\mathbf{h}_1, \cdots, \mathbf{h}_T]. \tag{2}$$

where T is the number of frames. Then we obtain $\mathbf{u}(X)$ by applying a unsupervised syllable segmentation model to get boundaries $[t_1, \cdots, t_{m(X)}]$, and mean pool within each segment:

$$\mathbf{u}_i(X) := \frac{1}{t_{i+1} - t_i} \sum_{t=t_i}^{t_{i+1}-1} \mathbf{h}_t, \; i = 1, \cdots, m(X). \tag{3}$$

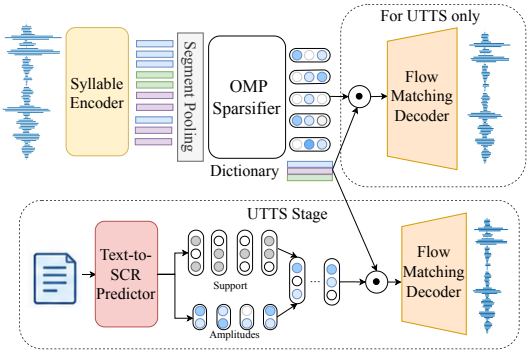


Figure 2: **Overview of the speech sparsifier.** A syllable-level encoder turns the waveform into frame-level features, then pools them by syllable boundaries. Then OMP produces the sparse codes. For UTTS, a flow-matching uses the codes and dictionary to reconstruct the waveform.

In other words, $\mathbf{u}(X)$ will be syllable-level segment features in this setting. Next, $\mathbf{u}(X)$ is sparsified into $\mathbf{z}(X)$ using sparse recovery techniques. In particular, we employ orthogonal matching pursuit (OMP) (Mallat & Zhang, 1993), given its strong theoretical guarantees with a fixed dictionary. To this end, OMP minimizes the following objective:

$$\mathbf{z}_i(X) = \underset{\mathbf{z} \in \mathbb{R}^K : \|\mathbf{z}\|_0 \le r}{\arg\min} \; \|\mathbf{\Phi}\mathbf{z} - \mathbf{u}_i(X)\|_F^2, \qquad i = 1, \cdots, m(X). \tag{4}$$

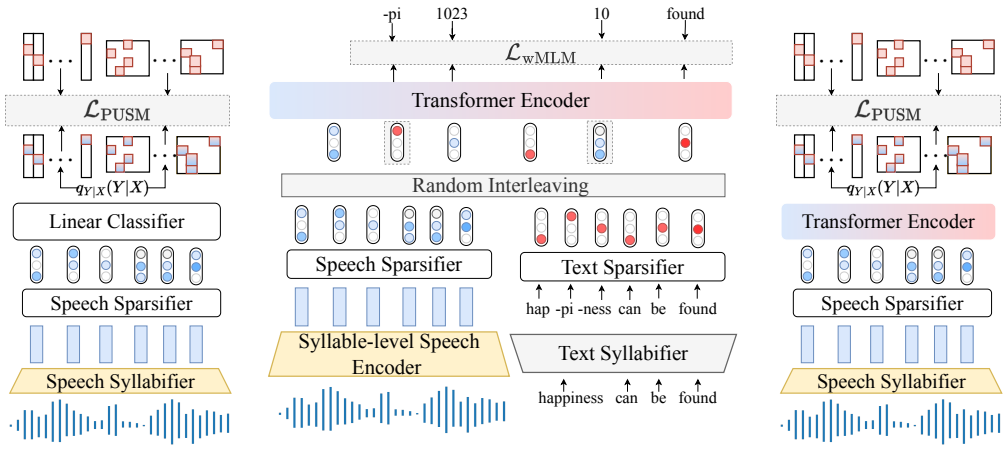
While optimize Eq. 4 exactly is NP-hard in general (Natarajan, 1995), OMP uses a greedy approach to optimize each sparse coefficients $z_{ik}(X)$'s sequentially. To obtain the dictionary Φ, we applied K-means on the segment-level features $\mathbf{u}_i(X)$'s and use the resulting centroids as the dictionary of atoms $\mathbf{\Phi}$. The current dictionary stays fixed during OMP training, since we found it already obtains good result, and we left more sophisticated dictionary learning approaches for future works.

3.4 Syllable-level speech sparsifier for UTTS

While Eq. 4 encourage the SCR to well approximate the original dense features, it can overly focus on semantic and lose acoustic details useful for generative tasks such as prosody and speaking style. Therefore, we additionally optimize the features to reconstruct the speech utterance for the UTTS task, as shown in Figure 2. To this end, we make the self-supervised speech encoder $\mathbf{h}_{\theta_e}(X)$ trainable with parameter θ_e, while using the same frozen syllable segmenter as in the UASR case to obtain the syllable-level representation $\mathbf{u}_{\theta_e}(X)$. We also apply OMP on $\mathbf{u}_{\theta_e}(X)$ to create a *sparsified* version of the representation $\hat{\mathbf{u}}_{\theta_e}(X) := \mathbf{\Phi}^* \mathbf{z}^*(X)$, where $(\mathbf{\Phi}^*, \mathbf{z}^*)$ is the optimal solution of OMP.

Further, we employ a speech generator to reconstruct the utterance X. Inspired by MatchaTTS (Mehta et al., 2024), the speech generator first predict syllable-level spectral means $\mathbf{M}$ from the sparsified syllable features $\mathbf{u}_{\theta_e}(X)$ using a *prior encoder* $\boldsymbol{\mu}_{\theta_g}$:

$$\forall i, \forall t \in [t_i, t_{i+1}), \qquad \mathbf{M}_t := \boldsymbol{\mu}_{\theta_g, i}(\mathbf{u}_{\theta_e}(X)), \qquad \widehat{\mathbf{M}}_t := \boldsymbol{\mu}_{\theta_g, i}(\hat{\mathbf{u}}_{\theta_e}(X))$$



(a) PUSM stage (b) Interleaving MLM stage (c) Second PUSM stage

Figure 3: **Overall architecture of SparCipher.**(a) A linear PUSM-based system is trained on SCR inputs. SparPUSM is the method with only this stage; (b) The interleaved MLM stage mixes predicted transcripts from SparPUSM with SCR tokens to perform joint speech-text language modeling with a transformer encoder; (c) A second PUSM loss is applied on the transformer encoder from (b).

where each spectral mean is repeated to the same length of each segment and the syllable durations t_i's can be predicted by a *duration predictor* $\mathbf{l}_{\theta_e}$ (Mehta et al., 2024) from $\mathbf{u}_{\theta_e}(X)$ during inference. To train the prior encoder and duration predictor, we uses two mean square error (MSE) losses with the mel-spectrogram X^{mel} of X and the log-durations as targets:

$$\mathcal{L}_{\text{prior}}(\theta_e, \theta_g) := \mathbb{E}_X \|\mathbf{M} - X^{\text{mel}}\|^2, \qquad \mathcal{L}_{\text{dur}}(\theta_e) := \mathbb{E}_X \|\mathbf{l}_{\theta_e,i}(\mathbf{u}_{\theta_e}(X)) - \log(t_{i+1} - t_i)\|^2. \quad (5)$$

Then it uses a *decoder* $\mathbf{v}_{\theta_g}$ to reconstruct the mel-spectrogram of X from the spectral means $\boldsymbol{\mu}(X)$ trained with the conditional flow matching objective (Lipman et al., 2023):

$$\mathcal{L}_{\text{CFM}}(\theta_e, \theta_g) := \mathbb{E}_{X,\tau} \|\mathbf{v}_{\theta_g}(X^{\text{mel},\tau}, \widetilde{\mathbf{M}}, \mathbf{s}_{\theta_e}(X), \tau) - (X^{\text{mel}} - X^{\text{mel},0})\|_F^2, \quad (6)$$

where τ is the flow time step, and $X^{\text{mel},\tau} \sim \mathcal{N}(\tau X^{\text{mel}}, (1-\tau)^2 I)$ is the noisy version of X^{mel} at flow time τ. $\mathbf{s}_{\theta_e}(X)$ denotes the speaker embedding extracted from an early layer of the encoder. In the first training stage, we set $\widetilde{\mathbf{M}} = \mathbf{M}$ to allow gradient flowing through the encoder; in the second stage, we instead set $\widetilde{\mathbf{M}} = \widehat{\mathbf{M}}$ to further adapt the decoder to sparsified features, which are used in the actual UTTS setup. A frozen HiFi-GAN vocoder (Kong et al., 2020) then converts the mel-spectrgram back to a speech waveform. The overall objective is then:

$$\mathcal{L}_{\text{SCR-TTS}}(\theta_e, \theta_g) = \mathcal{L}_{\text{CFM}}(\theta_e, \theta_g) + \lambda_1 \mathcal{L}_{\text{prior}}(\theta_e, \theta_g) + \lambda_2 \mathcal{L}_{\text{dur}}(\theta_e).$$

During UTTS, we train a *text-to-SCR predictor* $\hat{\mathbf{z}}_{\theta_{\text{TTS}}}(Y)$ before feeding the sparsified feature $\boldsymbol{\Phi}\hat{\mathbf{z}}_{\theta_{\text{TTS}}}(Y)$ into the speech generator, which is empirically more stable to train and less sensitive to label noise than direct text-to-dense feature prediction. By exploiting the sparse structure, the SCR prediction can be decomposed into two sub-tasks: *support prediction* and *amplitude prediction* of active atoms, where the former is a multilabel classification task while the latter is a low-dimensional regression task. To this end, we parameterize $\hat{\mathbf{z}}_{\theta_{\text{TTS}}}(Y) := \mathbf{b}(Y) \odot \mathbf{a}(Y)$, where $\odot$ denote elementwise multiplication, $\mathbf{b}(Y)$ denotes binary support and $\mathbf{a}(Y)$ denotes amplitudes, and use a combination of classification and regression objectives:

$$\mathcal{L}_{\text{UTTS}}(\theta_{\text{TTS}}) := \mathbb{E}_{X,\hat{Y}} \sum_{i=1}^{m(X)} \sum_{k=1}^{K} \mathbb{1}_{[z_{ik}(X)>0]} \left(-\log b_{ik}(\hat{Y}) + (a_{ik}(\hat{Y}) - z_{ik}(X))^2 \right) \quad (7)$$

where c_{ij} denotes the (i,j)-th entry of matrix $\mathbf{c}$, and $\hat{Y}$ is the pseudo-text transcript generated by the UASR. This decomposition also makes the model more efficient to train than directly using the dense features.

3.5 SCR-BASED DECIPHERMENT

To perform cross-modal decipherment, we develop **SparCipher**, an SCR-based UASR shown in Figure 3. Given the speech SCR $\mathbf{z} := \mathbf{z}(X)$ computed using Eq. 4, SparCipher first maps them to text tokens using positional unigram and skipgram matching (PUSM), which matches unimodal distributions across speech and text (Wang et al., 2024):

$$\mathcal{L}_{\text{PUSM}}(\mathbf{g}) = \sum_{i=1}^{m_{\max}} \|\mathbb{E}_X[\mathbf{g}_i(\mathbf{z})] - p_{Y_i}\|_1 + \sum_{r=1}^{R} \left\| \sum_{i=1}^{m_{\max}} \mathbb{E}_X[\mathbf{g}_i(\mathbf{z})\mathbf{g}_{i+r}(\mathbf{z})^\top] - p_{Y_i Y_{i+k}} \right\|_1, \quad (8)$$

where $m_{\max} = \max_{X\in\mathcal{X}} m(X)$ denotes the maximal syllable sequence length, R controls the maximal skip size and $\mathbf{g}(\cdot)$ denotes the ASR. Thanks to SCR's compatibility with text, we can feed sparse, continuous features instead of lossy discrete units as done in the original PUSM. In the initial stage of UASR training, we consider a simple linear+softmax ASR:

$$g_{\mathbf{w}_{\text{PUSM}}}(y_i|\mathbf{z}_i) := g_{\mathbf{w}_{\text{PUSM}},iy_i}(\mathbf{z}) = \frac{\exp(\mathbf{w}_{\text{PUSM},y_i}^\top \mathbf{z}_i)}{\sum_{y'} \exp(\mathbf{w}_{\text{PUSM},y'}^\top \mathbf{z}_i)}, \quad (9)$$

and refer to the approach as *SparPUSM*. This stage provides a warm start for the later decipherment stage with higher-order information, which are known to be slow to converge (Ni et al., 2025).

To capture higher-order crossmodal structure, we introduce a second stage based on *masked language modeling (MLM)*. Given pseudo transcripts from PUSM, we first randomly *interleave* PUSM-generated transcript tokens with syllable-level SCRs or discrete speech units to create sequences $\mathbf{o}$ or $\hat{\mathbf{o}}$ respectively as

$$\mathbf{o}_i := \begin{cases} \mathbf{z}_i, \text{ with prob. } 0.5 \\ \mathbf{e}_{\hat{Y}_i}, \text{ with prob. } 0.5 \end{cases}, \hat{\mathbf{o}}_i := \begin{cases} \mathbf{e}_{\arg\max_k z_{ik}}, \text{ with prob. } 0.5 \\ \mathbf{e}_{\hat{Y}_i}, \text{ with prob. } 0.5 \end{cases}, 1 \le i \le m(X).$$

where $\mathbf{e}_k$ denotes the k-th column of the identity matrix with proper size. Then we randomly sample a index mask $\mathcal{I} \subseteq \{1, \cdots, m(X)\}$, and train a transformer $g_{\text{MLM}}(\cdot)$ with separate embeddings for speech and text, an encoder $f_{\theta_{\text{MLM}}}$ and softmax prediction heads to predict the masked portion of the interleaved speech-text token sequence $\hat{\mathbf{o}}_{\mathcal{I}} := (\hat{\mathbf{o}}_i)_{i\in\mathcal{I}}$ from the unmasked portion of the interleaved SCR-text sequence $\mathbf{o}_{\mathcal{I}^c}$:

$$g_{\theta_{\text{MLM}}}(\hat{\mathbf{o}}_{\mathcal{I}}|\mathbf{o}_{\mathcal{I}^c}) := \prod_{i\in\mathcal{I}} \frac{\exp(\mathbf{w}_{\text{MLM},\hat{\mathbf{o}}_i}^\top f_i(\mathbf{o}_{\mathcal{I}^c}))}{\sum_k \exp(\mathbf{w}_{\text{MLM},k}^\top f_i(\mathbf{o}_{\mathcal{I}^c}))}, \quad (10)$$

which is then trained with a weighted MLM loss with random mixup (Ao et al., 2022; Ni et al., 2025):

$$\mathcal{L}_{\text{wMLM}}(\theta_{\text{MLM}}) := -\mathbb{E}_{\mathbf{o}} \log g_{\theta_{\text{MLM}}}(\hat{\mathbf{o}}_{\mathcal{I}}|\mathbf{o}_{\mathcal{I}^c}) g_{\theta_{\text{MLM}}}(\hat{\mathbf{o}}_{\mathcal{I}^c}|\mathbf{o}_{\mathcal{I}^c})^{\lambda_3}, \quad (11)$$

where λ_3 balances the role of $g_{\theta_{\text{MLM}}}$ as an MLM and as a cross-modal decoder. Finally, with the contextualized representation learned from $g_{\theta_{\text{MLM}}}$, we found it beneficial to have a *second* PUSM stage where $g_{\theta_{\text{MLM}}}(y_i|\mathbf{z})$'s is used in place of $g_{\mathbf{w}_{\text{PUSM}}}(y_i|\mathbf{z}_i)$. We called this model *SparCipher*.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Datasets. We evaluate SparPUSM and SparCipher on the 460-hour clean portion of LibriSpeech (Panayotov et al., 2015), 85-hour Mandarin AISHELL-3 (Shi et al., 2021), and 100-hour subsets of German, Dutch, and French MLS (Pratap et al., 2020; Tseng et al., 2024). We also consider two low-resource datasets: Cantonese MDCC (Yu et al., 2022) and Taigi SuiSiann (Ì-thuân, 2019). We follow the LibriSpeech split of Wang et al. (2026) and standard splits elsewhere. LibriSpeech includes a matched setting and several settings with mismatched speech and text distributions. To simulate domain shift, we replace 50% or 75% of the in-domain text with news or Wikipedia text. For UTTS, we train on the LibriSpeech UASR subset and evaluate on LJSpeech for single-speaker synthesis, following Ni et al. (2022), and on LibriSpeech for multi-speaker synthesis. For English and MLS, we compare linguistically informed Pyphen+ syllables with linguistically agnostic BPE+ units (Wang et al., 2026). Mandarin uses raw characters, Cantonese uses JyutPing, and Taigi uses an analogous syllabic notation. We retain the 1,024 most frequent syllables per language for multilingual UASR and the 2,048 most frequent tokens otherwise, mapping all remaining tokens to <OOV>.

Model	Student	Token	Matched CER (↓)	Unmatched CER (↓)
G2P-based approach				
wav2vec-U* (Baevski et al., 2021)	No	Phone	-	13.3
wav2vec-U 2.0* (Liu et al., 2023a)	No	Phone	-	12.2
REBORN* (Tseng et al., 2024)	No	Phone	-	8.3
G2P-free approach				
wav2vec-U (Baevski et al., 2021)	No	Char.	35.6	43.3
	Yes	Char.	33.8	42.1
REBORN (Tseng et al., 2024)	No	Char.	37.8	76.6
JSTTI (Ni et al., 2025)	No	Word	49.5	54.2
PUSM Wang et al. (2023c)	No	Syllable	35.5	57.7
	Yes	Syllable	33.0	54.7
SylCipher Wang et al. (2026)	No	Syllable	21.8	35.9
	Yes	Syllable	17.5	33.3
SparPUSM (ours)	No	Syllable	24.4	38.3
SparCipher (ours)	No	Syllable	**18.6**	**29.0**
	Yes	Syllable	**12.0**	**19.9**

Table 1: **UASR results on the LibriSpeech clean subsets.** *Student* indicates self-training with model-generated pseudo-labels. *Char.* denotes characters, and *Syllable* denotes tokens produced by the G2P-free Pyphen+ syllabifier. CER is character error rate. * indicates evaluation on dev-clean. We use a sparsity level of 16.

Model	Token	MLS			AISHELL-3		Low-resource	
		de	nl	fr	PER	CER	Cantonese	Taigi
wav2vec-U (Baevski et al., 2021)	Phone	32.5	40.2	39.8	74.9	–	75.7	69.0
wav2vec-U (Baevski et al., 2021)	Character	42.9	50.6	72.6	–	–	–	–
PUSM (Wang et al., 2023c)	Syllable	35.1	38.2	43.6	37.4	54.5	36.6	37.3
SylCipher Wang et al. (2026)	Syllable	32.3	35.6	41.7	29.9	49.2	35.0	35.5
SparPUSM (ours)	Syllable	31.5	**32.8**	41.8	34.0	45.3	35.7	**34.2**
SparCipher (ours)	Syllable	**29.4**	34.5	**35.8**	**28.2**	**35.5**	**31.9**	**34.2**

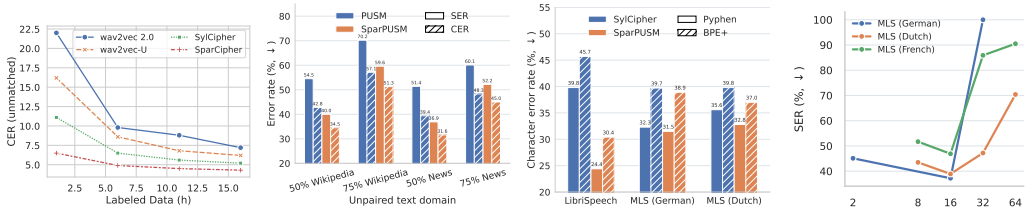
Table 2: **Multilingual and low-resource UASR results** (%, lower is better). German (de), Dutch (nl), and French (fr) share one multilingual MLS model; Mandarin (AISHELL-3), Cantonese (MDCC), and Taigi (SuiSiann) use separate models. The token column applies to MLS, Cantonese, and Taigi. Mandarin uses raw characters except for phone-based wav2vec-U. MLS is evaluated according to the listed token type, Cantonese and Taigi by phone error rate, and AISHELL-3 by PER and CER. Bold marks the best result in each column. Sparsity is 16 for MLS, AISHELL-3, and Cantonese, and 1, with continuous amplitudes, for Taigi.

Evaluation metrics. For UASR, we report syllable (SER), phone (PER), or character error rate (CER), depending on the output units. For UTTS, we report word error rate (WER) and CER for intelligibility and speaker similarity for voice consistency.

Model configurations. We use Sylber (Cho et al., 2025) for syllable segmentation and XEUS (Chen et al., 2024) as the frame-level speech encoder. For UTTS, XEUS layer 6 also provides speaker embeddings. The SCR dictionary has one atom per text-vocabulary item. We vary the number of active atoms and use 16 by default; lower values work better with BPE tokens and for low-resource Taigi. SparCipher uses the two-layer Transformer and most MLM hyperparameters of SylCipher (Wang et al., 2026); its PUSM stages follow the original PUSM settings. For MLS, the interleaving MLM is trained jointly on all three languages with language-ID prefixes, whereas the PUSM stages are language-specific. At inference, we suppress <OOV> and emit the closest retained

Transcript Model	Model	LibriSpeech		LJSpeech	
		WER (↓)	SSIM (↑)	WER (↓)	CER (↓)
G2P-free supervised approach					
Groundtruth	Chen et al. (2025)	2.42*	0.66*	2.93†	-
Groundtruth	Dense+CFM	9.1	0.49	-	-
Groundtruth	RVQ+CFM	17.9	0.46	-	-
G2P-based unsupervised approach					
wav2vec-U (Baevski et al., 2021)	Ni et al. (2022)	-	-	12.0	4.6
G2P-free unsupervised approach					
wav2vec-U (Baevski et al., 2021)	Tacotron 2	-	-	53.3	32.5
SylCipher Wang et al. (2026)	Dense+CFM	200	0.20	-	-
SylCipher Wang et al. (2026)	RVQ+CFM	41.1	**0.44**	-	-
SparCipher (ours)	RVQ+CFM	34.6	0.39	36.8	27.1
SparCipher (ours)	SCR+CFM	**33.1**	0.40	**34.1**	**25.9**

Table 3: **UTTS results on LibriSpeech and LJSpeech.** We report WER and CER (lower is better) and, for LibriSpeech, speaker similarity (SSIM; higher is better). Dense denotes continuous XEUS features (Chen et al., 2024); RVQ uses eight residual quantization layers. * indicates training on the larger Emilia dataset, and † indicates a different data split (He et al., 2024).



(a) Semi-supervised ASR versus labeled data (b) Effect of unpaired-text domain shift (c) Effect of text tokenization (d) Effect of SCR sparsity

Figure 4: **Ablation analysis.** We study (a) the amount of labeled speech, (b) domain shift in unpaired text, (c) text tokenization, and (d) SCR sparsity. SER denotes syllable error rate; lower is better.

syllable. The predicted transcript is further refined via self-training (Chen et al., 2019) following similar setup in (Wang et al., 2026).

For UTTS, the text-to-SCR predictor follows the Matcha-TTS text-decoder backbone (Mehta et al., 2024), replacing its mel prediction head with separate SCR support and amplitude heads. The prior encoder replaces Matcha-TTS's text embedding with a projection of syllable-level speech embeddings, and the duration predictor omits monotonic alignment. A Matcha-TTS-style conditional flow-matching decoder maps aligned SCRs to mel-spectrograms using a 1D U-Net with convolutional residual and Transformer blocks; a vocoder then generates waveforms. Unlike the autoregressive TTS system conditioned on low-dimensional dense tokens in Sylber 2.0 (Cho et al., 2026), our system operates non-autoregressively and conditions on SCRs since we found the former setting to be unstable with noisy syllable-level transcripts. Additional details are provided in Appendix A.

Baselines. UASR baselines include JSTTI (Ni et al., 2025); phone- and character-level wav2vec-U (Baevski et al., 2021); phone-level REBORN (Tseng et al., 2024); and syllable-level PUSM (Wang et al., 2023c) and SylCipher (Wang et al., 2026). For UTTS, we reproduce the Tacotron 2 setup (Shen et al., 2018) of Ni et al. (2022). We also replace SCRs with either dense features or eight-layer RVQ units while keeping the rest of the model fixed. Each RVQ codebook contains 2,048 K-means centroids. Finally, we train the same UTTS model on pseudo-transcripts from SparCipher, SylCipher, and wav2vec-U.

4.2 Main Results

Table 1 reports LibriSpeech UASR results with matched and unmatched text. SparCipher is the strongest G2P-free system in both settings. Without self-training, it achieves 18.6% matched and 29.0% unmatched CER, relative improvements of 14.7% and 19.2% over SylCipher. Self-training lowers these errors to 12.0% and 19.9%, improving over self-trained SylCipher by 31.4% and 40.2%. SparPUSM likewise improves over PUSM, reducing CER from 35.5% to 24.4% with matched text and from 57.7% to 38.3% with unmatched text. SparCipher also has a smaller domain-mismatch gap than SylCipher after self-training (7.9 versus 15.8 points). Thus, sparsity improves both alignment methods and complements self-training.

Table 2 extends the evaluation to six languages. On MLS, SparPUSM outperforms PUSM in German, Dutch, and French, while SparCipher improves over SylCipher in all three languages. SparCipher is best on German (29.4%) and French (35.8%), and SparPUSM is best on Dutch (32.8%). The largest gain is on French, where SparCipher reduces error by 14.1% relative to SylCipher. These consistent improvements show that sparse representations generalize beyond English. On AISHELL-3, SparCipher achieves the best PER (28.2%) and CER (35.5%), improving over SylCipher by 5.7% and 27.8% relative, respectively. SparPUSM also improves over PUSM, from 37.4% to 34.0% PER and from 54.5% to 45.3% CER. The larger CER gains suggest that sparse representations better capture Mandarin character structure. In the low-resource settings, SparCipher is best on Cantonese (31.9%), while SparCipher and SparPUSM tie on Taigi (34.2%). Relative to SylCipher, SparCipher lowers error from 35.0% to 31.9% on Cantonese and from 35.5% to 34.2% on Taigi. SparPUSM similarly improves over PUSM on both languages. These results confirm that sparsity remains useful with limited training data.

Table 3 evaluates UASR transcripts for downstream UTTS. Direct prediction of dense features is unstable: SylCipher with Dense+CFM reaches 200% WER and 0.20 SSIM on LibriSpeech. With the same RVQ+CFM synthesizer, SparCipher reduces SylCipher's WER from 41.1% to 34.6%, although SSIM falls from 0.44 to 0.39. This decline in SSIM may be due to the better syllable boundaries produced by the JE2E step of SylCipher, which is omitted in SparCipher for simplicity. Better boundary refinement methods will be left as future works. Using SCR+CFM further improves WER to 33.1% and SSIM to 0.40. On LJSpeech, SparCipher with RVQ+CFM reduces the wav2vec-U-Tacotron 2 baseline from 53.3% to 36.8% WER and from 32.5% to 27.1% CER. SCR+CFM performs best, with 34.1% WER and 25.9% CER. Thus, SCRs provide a more effective interface for synthesis than either dense features or RVQ units, although a substantial gap from supervised TTS remains. Audio samples by our SCR-based UTTS are provided in Appendix B.

4.3 Ablation Analysis

Figure 4 summarizes four ablations. First, SparCipher gives the lowest semi-supervised ASR error at every labeled-data budget, with the largest advantage in the one-hour setting. Second, domain mismatch degrades all systems, but SparPUSM consistently outperforms PUSM across text sources and mismatch levels. Third, Pyphen+ outperforms BPE+ for both SylCipher and SparPUSM on LibriSpeech, German, and Dutch, while SparPUSM remains better under either tokenizer. Finally, a sparsity level of 16 performs best across the MLS languages; larger values sharply increase error. Together, these results show that SCRs are robust to limited supervision, domain shift, and tokenization choice, but require an appropriate sparsity level.

5 Conclusions

We proposed SparPUSM and SparCipher, which achieve state-of-the-art results on G2P-free UASR, outperforming prior work by up to 17% across several languages. Building on SparCipher, we take a first step toward unpaired text-to-speech generation, showing consistent gains over prior UTTS baselines. Our results suggest SCRs offer a better capacity-compatibility tradeoff than both discrete or dense continuous representations and improve generalization in unpaired speech-text learning tasks. Next, we would like to explore end-to-end training with an SAE and broaden our evaluation to more languages and unpaired speech–text learning tasks.

AI USE STATEMENT

In this work, we used generative AI tools for refining hypotheses, providing feedback on research methodology, and assisting in implementing parts of the code/method. We have not used generative AI for conceptualizing research ideas, initial drafting, experimental design, thematic data analysis, or interpreting results. All other standard disclosure tasks are not applicable. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Michal Aharon, Michael Elad, and Alfred M. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54 (11):4311–4322, 2006. doi: 10.1109/TSP.2006.881199.

Junyi Ao, Rui Wang, Long Zhou, Chengyi Wang, Shuo Ren, Yu Wu, Shujie Liu, Tom Ko, Qing Li, Yu Zhang, Zhihua Wei, Yao Qian, Jinyu Li, and Furu Wei. SpeechT5: Unified-modal encoder-decoder pre-training for spoken language processing. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 5723–5738. Association for Computational Linguistics, 2022. doi: 10.18653/V1/2022.ACL-LONG.393. URL https://doi.org/10.18653/v1/2022.acl-long.393.

Georgii Aparin, Tasnima Sadekova, Alexey Rukhovich, Assel Yermekova, Laida Kushnareva, Vadim Popov, Kristian Kuznetsov, and Irina Piontkovskaya. Audiosae: Towards understanding of audio-processing models with sparse autoencoders. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3221–3254. Association for Computational Linguistics, 2026.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, pp. 12449–12460, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92d1e1eb1cd6f9fba3227870bb6d7f07-Abstract.html.

Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. Unsupervised speech recognition. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 27826–27839, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/ea159dc9788ffac311592613b7f71fbb-Abstract.html.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nicholas L. Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E. Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-yi Lee, and Lin-Shan Lee. Completely unsupervised phoneme recognition by a generative adversarial network harmonized with iteratively refined hidden markov models. In Gernot Kubin and Zdravko Kacic (eds.), *Interspeech 2019, 20th Annual Conference of the International Speech Communication Association, Graz, Austria, 15-19 September 2019*, pp. 1856–1860. ISCA, 2019. doi: 10.21437/INTERSPEECH.2019-2068. URL https://doi.org/10.21437/Interspeech.2019-2068.

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,

Jian Wu, Michael Zeng, Xiangzhan Yu, and Furu Wei. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Signal Process.*, 16(6):1505–1518, 2022. doi: 10.1109/JSTSP.2022.3188113. URL https://doi.org/10.1109/JSTSP.2022.3188113.

Scott Shaobing Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33–61, 1998. doi: 10.1137/S1064827596304010.

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 *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 10205–10224. Association for Computational Linguistics, 2024. doi: 10.48550/arXiv.2407.00837. Preprint available on arXiv: arXiv:2407.00837.

Yushen Chen, Zhikang Niu, Ziyang Ma, Keqi Deng, Chunhui Wang, Jian Zhao, Kai Yu, and Xie Chen. F5-TTS: A fairytaler that fakes fluent and faithful speech with flow matching. In *ACL*, pp. 6255–6271, 2025. doi: 10.18653/v1/2025.acl-long.313. URL https://aclanthology.org/2025.acl-long.313/.

Cheol Jun Cho, Nicholas Lee, Akshat Gupta, Dhruv Agarwal, Ethan Chen, Alan W. Black, and Gopala K. Anumanchipalli. Sylber: Syllabic embedding representation of speech from raw audio. In *ICLR*, Singapore, 2025.

Cheol Jun Cho, Nayeon Lee, Alan W Black, and Gopala Anumanchipalli. Sylber 2.0: A universal syllable embedding. *arXiv preprint arXiv:2601.22306*, 2026.

Ikhyun Cho and Julia Hockenmaier. Toward efficient sparse autoencoder-guided steering for improved in-context learning in large language models. In *EMNLP*, pp. 28961–28973, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1474. URL https://aclanthology.org/2025.emnlp-main.1474/.

Ha-Yeong Choi, Sang-Hoon Lee, and Seong-Whan Lee. Diff-HierVC: Diffusion-based Hierarchical Voice Conversion with Robust Pitch Generation and Masked Prior for Zero-shot Speaker Adaptation. In *Proc. INTERSPEECH 2023*, pp. 2283–2287, 2023. doi: 10.21437/Interspeech.2023-817.

Kwanghee Choi, Eunjung Yeo, Cheol Jun Cho, David Harwath, and David R. Mortensen. [b] = [d] - [t] + [p]: Self-supervised speech models discover phonological vector arithmetic. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 11048–11069, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.537. URL https://aclanthology.org/2026.findings-acl.537/.

Hoagy Cunningham, Aidan Ewart, Logan Riggs Smith, Robert Huben, and Lee Sharkey. Sparse autoencoders find highly interpretable features in language models. In *ICLR*, 2024.

Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi. High fidelity neural audio compression. *Transactions on Machine Learning Research*, 2023. URL https://openreview.net/forum?id=ivCd8z8zR2.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable LLM feature circuits. In *Neural Information Processing Systems*, volume 37, 2024. URL https://papers.nips.cc/paper_files/paper/2024/hash/2b8f4db0464cc5b6e9d5e6bea4b9f308-Abstract-Conference.html.

Leo Gao, Tom Dupré la Tour, Henk Tillman, Gabriel Goh, Rajan Troll, Alec Radford, Ilya Sutskever, Jan Leike, and Jeffrey Wu. Scaling and evaluating sparse autoencoders. In *ICLR*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/42ef3308c230942d223c411adf182c88-Abstract-Conference.html.

James Glass. Towards unsupervised speech processing. In *International Conference on Information Sciences, Signal Processing and their Applications*, 2012. URL https://groups.csail.mit.edu/sls/publications/2012/Glass-ISSPA12.pdf.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Neural Information Processing Systems*, volume 27, pp. 2672–2680, 2014.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *ICML*, pp. 399–406, 2010. URL https://mlanthology.org/icml/2010/gregor2010icml-learning/.

Difei Gu, Yunhe Gao, Gerasimos Chatzoudis, Zihan Dong, Guoning Zhang, Bangwei Guo, Yang Zhou, Mu Zhou, and Dimitris Metaxas. LUCID-SAE: Learning unified vision-language sparse codes for interpretable concept discovery, 2026. URL https://arxiv.org/abs/2602.07311.

Xingwen Han and Hao Tang. Identifying the minimal and maximal phonetic subspace of speech representations. In *ICASSP*, pp. 15387–15391, 2026. doi: 10.1109/ICASSP55912.2026.11463314.

Haorui He, Zengqiang Shang, Chaoren Wang, Xuyuan Li, Yicheng Gu, Hua Hua, Liwei Liu, Chen Yang, Jiaqi Li, Peiyang Shi, Yuancheng Wang, Kai Chen, Pengyuan Zhang, and Zhizheng Wu. Emilia: An extensive, multilingual, and diverse speech dataset for large-scale speech generation. In *2024 IEEE Spoken Language Technology Workshop (SLT)*, pp. 885–890, 2024. doi: 10.1109/SLT61566.2024.10832365.

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 Trans. Audio Speech Lang. Process.*, 29: 3451–3460, 2021. doi: 10.1109/TASLP.2021.3122291. URL https://doi.org/10.1109/TASLP.2021.3122291.

Adam Karvonen, Can Rager, Johnny Lin, Curt Tigges, Joseph Isaac Bloom, David Chanin, Yeu-Tong Lau, Eoin Farrell, Callum Stuart Mcdougall, Kola Ayonrinde, Demian Till, Matthew Wearden, Arthur Conmy, Samuel Marks, and Neel Nanda. SAEBench: A comprehensive benchmark for sparse autoencoders in language model interpretability. In *ICML*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29223–29264, 2025. URL https://proceedings.mlr.press/v267/karvonen25a.html.

Chiraag Kaushik, Davis Barch, and Andrea Fanelli. Learning multimodal dictionary decompositions with group-sparse autoencoders. In *ICLR*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/32d51148004068b38d9f70c457541880-Abstract-Conference.html.

Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. HiFi-GAN: Generative adversarial networks for efficient and high fidelity speech synthesis. In *Neural Information Processing Systems*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/c5d736809766d46260d816d8dbc9eb44-Abstract.html.

Sébastien Lachapelle, Pau Rodríguez, Yash Sharma, Katie E. Everett, Rémi Le Priol, Alexandre Lacoste, and Simon Lacoste-Julien. Disentanglement via mechanism sparsity regularization: A new principle for nonlinear ICA. In *Proceedings of the First Conference on Causal Learning and Reasoning*, volume 177 of *Proceedings of Machine Learning Research*, pp. 428–484. PMLR, 2022.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Alexander H. Liu, Cheng-I Jeff Lai, Wei-Ning Hsu, Michael Auli, Alexei Baevski, and James Glass. Simple and effective unsupervised speech synthesis. In *Interspeech 2022*, pp. 843–847. ISCA, 2022. doi: 10.21437/Interspeech.2022-11071.

Alexander H. Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. Towards end-to-end unsupervised speech recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*, pp. 221–228, 2023a. doi: 10.1109/SLT54892.2023.10023187.

Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin shan Lee. Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings. In *Interspeech*, 2018. URL https://www.isca-speech.org/archive_v0/Interspeech_2018/pdfs/1800.pdf.

Oli Danyi Liu, Hao Tang, and Sharon Goldwater. Self-supervised predictive coding models encode speaker and phonetic information in orthogonal subspaces. In *Interspeech*, 2023b.

Julien Mairal, Francis Bach, and Jean Ponce. Task-driven dictionary learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(4):791–804, 2012. doi: 10.1109/TPAMI.2011.156.

Alireza Makhzani and Brendan J. Frey. k-sparse autoencoders. In *ICLR*, 2014. URL https://arxiv.org/abs/1312.5663.

Alireza Makhzani and Brendan J. Frey. Winner-take-all autoencoders. In *Neural Information Processing Systems*, volume 28, 2015. URL https://papers.nips.cc/paper_files/paper/2015/hash/5129a5ddcd0dcd755232baa04c231698-Abstract.html.

Stéphane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993. doi: 10.1109/78.258082.

Shivam Mehta, Ruibo Tu, Jonas Beskow, Éva Székely, and Gustav Eje Henter. Matcha-TTS: A fast TTS architecture with conditional flow matching. In *ICASSP*, 2024.

Videet Mehta, Liming Wang, Hilde Kuehne, Rogerio Feris, James R. Glass, and M. Jehanzeb Mirza. CALM: Class-conditional sparse attention vectors for large audio-language models. In *EMNLP*, 2026.

Chancharik Mitra, Brandon Huang, Tianning Chai, Zhiqiu Lin, Assaf Arbelle, Rogerio Feris, Leonid Karlinsky, Trevor Darrell, Deva Ramanan, and Roei Herzig. Enhancing few-shot vision-language classification with large multimodal model features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2025.

B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24 (2):227–234, 1995. doi: 10.1137/S0097539792240406.

Junrui Ni, Liming Wang, Heting Gao, Kaizhi Qian, Yang Zhang, Shiyu Chang, and Mark Hasegawa-Johnson. Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition. In Hanseok Ko and John H. L. Hansen (eds.), *Interspeech 2022, 23rd Annual Conference of the International Speech Communication Association, Incheon, Korea, 18-22 September 2022*, pp. 461–465. ISCA, 2022. doi: 10.21437/INTERSPEECH.2022-816. URL https://doi.org/10.21437/Interspeech.2022-816.

Junrui Ni, Liming Wang, Yang Zhang, Kaizhi Qian, Heting Gao, Mark Hasegawa-Johnson, and Chang D. Yoo. Towards unsupervised speech recognition without pronunciation models. *arXiv*, 2025. URL https://arxiv.org/pdf/2406.08380.

Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. doi: 10.1038/381607a0.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: An asr corpus based on public domain audio books. In *ICASSP*, pp. 5206–5210. IEEE, 2015. doi: 10.1109/ICASSP.2015.7178964.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. Mls: A large-scale multilingual dataset for speech research. *ArXiv*, abs/2012.03411, 2020.

Senthooran Rajamanoharan, Arthur Conmy, Lewis Smith, Tom Lieberum, Vikrant Varma, János Kramár, Rohin Shah, and Neel Nanda. Improving sparse decomposition of language model activations with gated sparse autoencoders. In *Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/01772a8b0420baec00c4d59fe2fbace6-Abstract-Conference.html.

Marc'Aurelio Ranzato, Christopher Poultney, Sumit Chopra, and Yann LeCun. Efficient learning of sparse representations with an energy-based model. In *Neural Information Processing Systems*, volume 19, 2006. URL https://papers.nips.cc/paper_files/paper/2006/hash/87f4d79e36d68c3031ccf6c55e9bbd39-Abstract.html.

Jason Rolfe and Yann LeCun. Discriminative recurrent sparse auto-encoders. In *International Conference on Learning Representations (ICLR)*, 2013.

Jonathan Shen, Ruoming Pang, Ron J. Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, R. J. Skerry-Ryan, Rif A. Saurous, Yannis Agiomyrgiannakis, and Yonghui Wu. Natural TTS synthesis by conditioning WaveNet on mel spectrogram predictions. In *ICASSP*, pp. 4779–4783, 2018. doi: 10.1109/ICASSP.2018.8461368.

Jiatong Shi, Chan-Jan Hsu, Holam Chung, Dongji Gao, Paola Garcia, Shinji Watanabe, Ann Lee, and Hung yi Lee. Bridging speech and text pre-trained models with unsupervised asr. In *ICASSP*, pp. 1–5. IEEE, 2023. doi: 10.1109/ICASSP49357.2023.10096827.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. Aishell-3: A multi-speaker mandarin tts corpus. In *Interspeech 2021*, pp. 2756–2760, 2021. doi: 10.21437/Interspeech.2021-755.

Pablo Sprechmann, Alexander M. Bronstein, and Guillermo Sapiro. Learning efficient sparse and low rank models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1821–1833, 2015. doi: 10.1109/TPAMI.2015.2392758.

Hillel Sreter and Raja Giryes. Learned convolutional sparse coding. In *ICASSP*, 2018. URL https://arxiv.org/abs/1711.00328.

Francesco Tonolini, Bjørn Sand Jensen, and Roderick Murray-Smith. Variational sparse coding. In Ryan P. Adams and Vibhav Gogate (eds.), *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, volume 115 of *Proceedings of Machine Learning Research*, pp. 690–700. PMLR, 22–25 Jul 2020. URL https://proceedings.mlr.press/v115/tonolini20a.html.

Liang-Hsuan Tseng, En-Pei Hu, David Cheng-Han Chiang, Yuan Tseng, Hung-yi Lee, Lin-Shan Lee, and Shao-Hua Sun. REBORN: reinforcement-learned boundary segmentation with iterative training for unsupervised ASR. *CoRR*, abs/2402.03988, 2024. doi: 10.48550/ARXIV.2402.03988. URL https://doi.org/10.48550/arXiv.2402.03988.

Changhan Wang, Hirofumi Inaguma, Peng-Jen Chen, Ilia Kulikov, Yun Tang, Wei-Ning Hsu, Michael Auli, and Juan Pino. Simple and effective unsupervised speech translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10771–10784, Toronto, Canada, July 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.602. URL https://aclanthology.org/2023.acl-long.602/.

Liming Wang, Mark Hasegawa-Johnson, and Chang Yoo. A theory of unsupervised speech recognition. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1192–1215, Toronto, Canada, July 2023b. Association for Computational Linguistics.

Liming Wang, Junrui Ni, Heting Gao, Jialu Li, Kai Chieh Chang, Xulin Fan, Junkai Wu, Mark Hasegawa-Johnson, and Chang Dong Yoo. Listen, decipher and sign: Toward unsupervised speech-to-sign language recognition. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pp. 6785–6800. Association for Computational Linguistics, 2023c. doi: 10.18653/V1/2023.FINDINGS-ACL.424. URL https://doi.org/10.18653/v1/2023.findings-acl.424.

Liming Wang, Mark Hasegawa-Johnson, and Chang D. Yoo. Unsupervised speech recognition with n-skipgram and positional unigram matching. In *ICASSP*, pp. 10936–10940. IEEE, 2024. doi: 10.1109/ICASSP48485.2024.10446327. URL https://doi.org/10.1109/ICASSP48485.2024.10446327.

Liming Wang, Kai-Wei Chang, Kunio Kashino, David Harwath, Mark Hasegawa-Johnson, and James R. Glass. Unsupervised speech recognition at the syllable level. In *EMNLP*, 2026.

Dongchao Yang, Songxiang Liu, Rongjie Huang, Jinchuan Tian, Chao Weng, and Yuexian Zou. Hifi-codec: Group-residual vector quantization for high fidelity audio codec, 2023.

Tiezheng Yu, Rita Frieske, Peng Xu, Samuel Cahyawijaya, Cheuk Tung Shadow Yiu, Holy Lovenia, Wenliang Dai, Elham J. Barezi, Qifeng Chen, Xiaojuan Ma, Bertram E. Shi, and Pascale Fung. Automatic speech recognition datasets in cantonese: A survey and new dataset. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, 2022. URL `https://aclanthology.org/2022.lrec-1.696/`.

Zeyu Yun, Yubei Chen, Bruno A. Olshausen, and Yann LeCun. Transformer visualization via dictionary learning: Contextualized embedding as a linear superposition of transformer factors. In *Proceedings of the Second Workshop on Deep Learning inside Out: The What, Why, and How of Machine Learning Policies (DeeLIO)*, pp. 12–26. Association for Computational Linguistics, 2021. doi: 10.18653/v1/2021.deelio-1.2.

Neil Zeghidour, Alejandro Luebs, Ahmed Omran, Jan Skoglund, and Marco Tagliasacchi. Soundstream: An end-to-end neural audio codec. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 30:495–507, 2022. doi: 10.1109/TASLP.2021.3129994.

Ì-thuân. Suísiann Dataset. `https://suisiann-dataset.ithuan.tw/`, 2019. Accessed: 2026-01-22.

A Implementation Details

SparCipher The shared encoder, pre-nets and post-nets following SylCipher (Wang et al., 2026), which is in term following JSTTI (Ni et al., 2025). Unlike the previous two approaches, we instead remove the differentiable boundary detector (soft-pooler) for simplicity. The detailed hyperparameter settings can be found in Table 6 of (Wang et al., 2026). During the interleaving MLM training, we use equal probabilities for speech or text tokens at each position of the interleaving sequence. We use an Adam optimizer with a learning rate 2×10^{-4}, $\beta = (0.9, 0.98)$ and $\epsilon = 10^{-6}$ with no weight decay and gradient-norm clipping of 20. We also use polynomial learning rate decay (power 1) and 150 warmup updates Poisson span masking with mean length 3.5 and a mask budget below 30%. Replace 90% of selected positions by `MASK` and 10% by a random token. Masked-token loss weight 1; unmasked token loss weight 0.5; speech and text modalities weighted equally. We train the model with a batch of no more than 500 tokens and an update frequencies of 16 until the validation SER flattens, which takes around 15-30k steps.

For the PUSM stages, we use an update frequency of 7200 to approximately use the whole dataset as a single batch; Codebook mix-up probability is set to 0. Each PUSM stage is trained for a maximum of 2000 epochs (or 2000 steps by using the whole dataset as a single batch) and stop when paired validation SER flattens. For the first linear PUSM stage, we use an Adam optimizer with the same hyperparameter settings as the MLM stage but with a learning rate of 0.4; for the second PUSM stage, a learning rate of 0.001 is used.

For pseudo-label self-training, we use a XEUS encoder as the student with a one-layer bidirectional LSTM with 512 hidden units as the classifier head. We use an Adam optimizer with a learning rate of 10^{-4}, $\beta = (0.9, 0.98)$, $\epsilon = 10^{-8}$, and a batch size of 32 for 5 epochs and we select the best checkpoint using validation CER. The semi-supervised setup follows the self-training setup, but with an additional stage where the model is finetuned only on the paired data for 100 epochs.

All UASR experiments are done on two NVIDIA RTX A6000 GPUs.

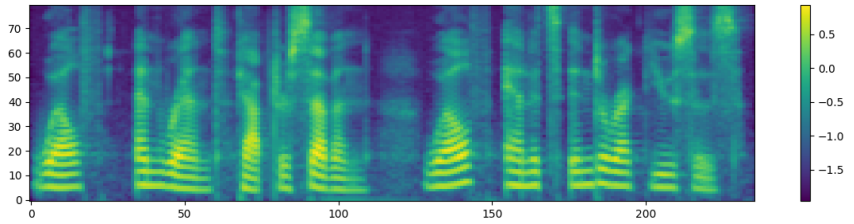
SCR-based UTTS Our UTTS is modified from Matcha-TTS (Mehta et al., 2024). To make the frame rate easily convertible with self-supervised features, the mel spectrogram uses a sampling rate of 16kHz, frame size of 320, window size of 1280, and 1280 FFT bins. This results in a 50Hz frame rate. We also choose a pretrained HiFiGAN vocoder from Diff-HierVC (Choi et al., 2023) that supports this mel spectrogram setting[1].

[1] https://github.com/hayeong0/Diff-HierVC

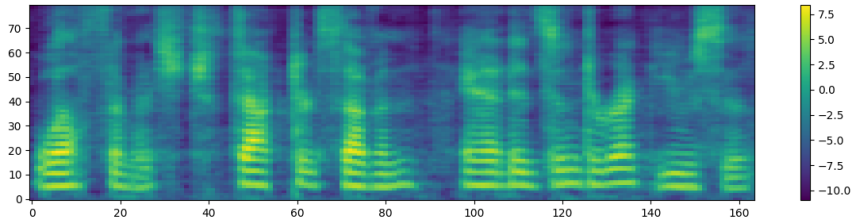
For the trainable SCR encoder, we train only the last three transformer layers. For the OMP, we set the audio dimension to be 1024, the number of atoms to be 2048 and the sparsity level to be 16. For both the text-to-SCR and the prior encoders, we use six transformer layers with 1024 encoder width, 1024 feedforward width, RoPE positional encoding, and 4 attention heads. A dropout of 0.1 is also used during training. For the duration predictor, we use a single convolutional layer on top of the prior encoder (before the Mel-space conversion head) with filter channels of 256, a kernel size of 3, and a dropout of 0.1.

For the flow-matching decoder, we use a 1D transformer U-Net decoder with two up and down blocks each with skip connections and a kernel size of $[256, 256]$. Six ResNet-plus-Transformer middle blocks with two transformer blocks after each ResNet block are used in total. In each transformer block, four attention heads with 64 dimensions each are used. GELU nonlinearlity is used for the MLP inside the transformer. Training uses a dropout of 0.05. Sinusoidal flow time embedding with 1024 dimensions is used. The ResNet convolutions use a kernel size of 3, GroupNorm with 8 groups and Mish activation. A final 1×1 convolutions is used to convert to the output channels.

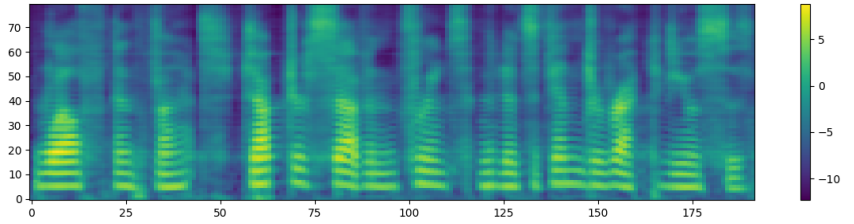
For training the UTTS, we set the loss weights to be 1.0 each for support, coefficient, prior, duration and flow matching losses. A batch size of 8 per GPU, gradient accumulation 2, and two NVIDIA RTX A6000 GPUs (nominal effective batch size 32, assuming distributed training uses both) are used. The Adam optimizer is used with a learning rate of 10^{-4}, 500 warmup steps, and 1M total/max steps.



(a) Reference

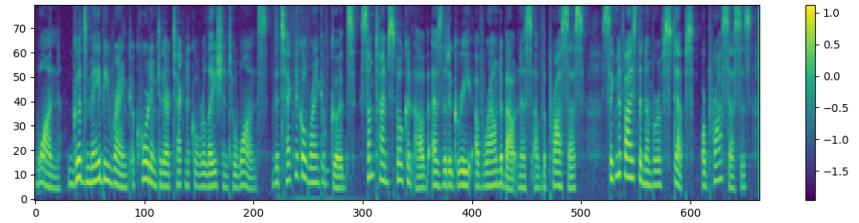


(b) RVQ

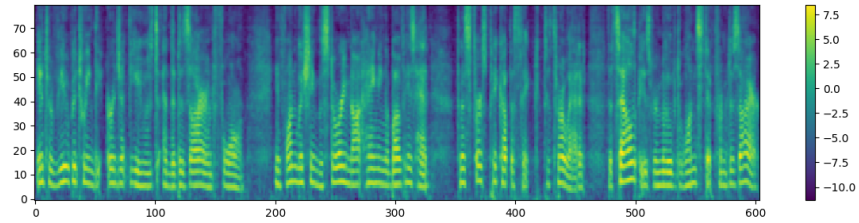


(c) SCR

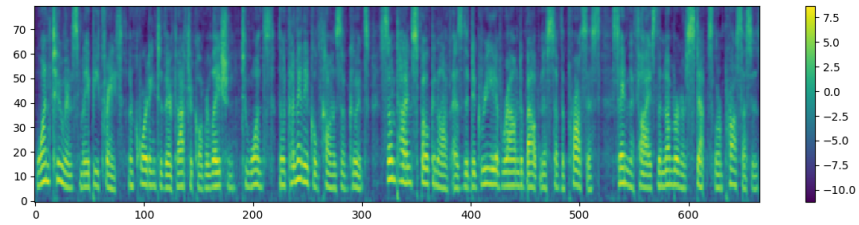
Figure 5: **Mel-spectrogram comparison for example 1.** Reference recording and speech synthesized from RVQ and sparse continuous representations (SCRs) for "wealth and rent chapter seven wealth and its indirect uses".



(a) Reference (example 2)



(b) RVQ (example 2)

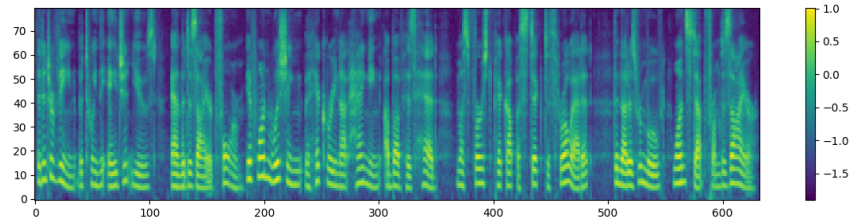


(c) SCR (example 2)

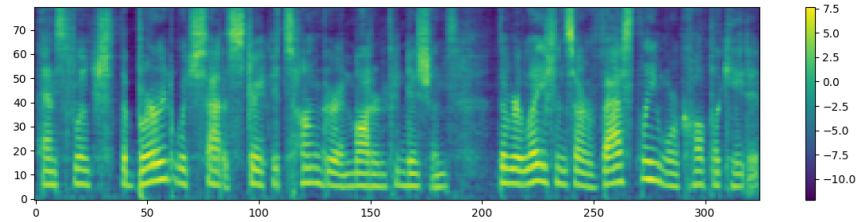
Figure 6: **Mel-spectrogram comparison for example 2.** Reference recording and speech synthesized from RVQ and sparse continuous representations (SCRs) for "wealth and rent chap ter sev en wealth and its in di rect us es".

B UTTS EXAMPLES

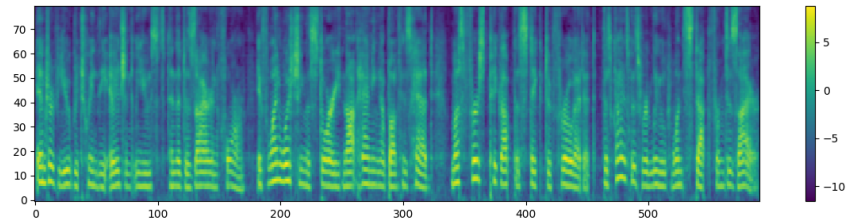
Figures 5-8 compare reference mel-spectrograms with outputs from the RVQ- and SCR-based UTTS systems in Table 3. Additional audio samples are provided in the Supplementary Material.



(a) Reference (example 3)

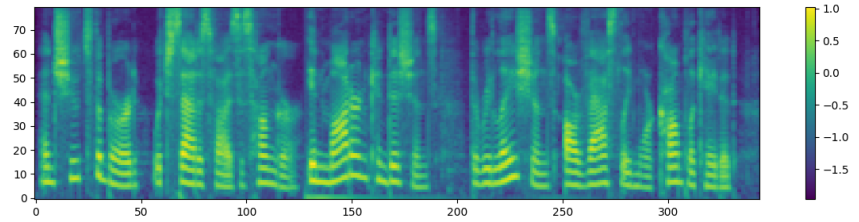


(b) RVQ (example 3)

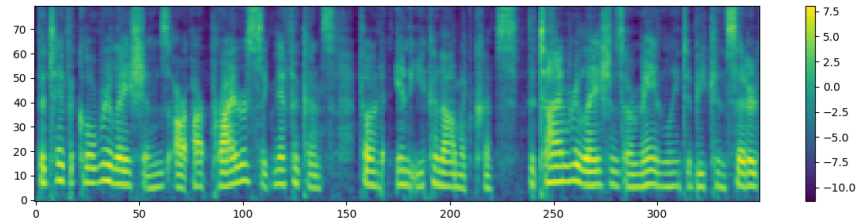


(c) SCR (example 3)

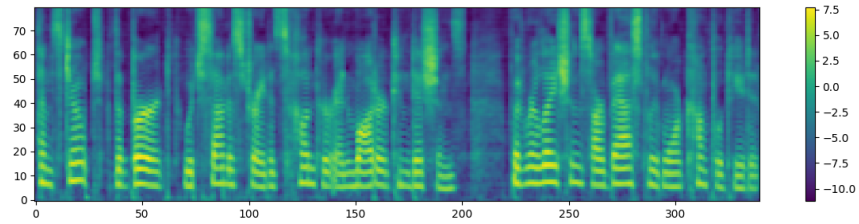
Figure 7: **Mel-spectrogram comparison for example 3.** Reference recording and speech synthesized from RVQ and sparse continuous representations (SCRs) for “that in ter vene be tween the agent used and the de sired form if one wish ing the hick o ry nut hang ing above his head must first pick up a stick to throw at it the nut is re moved one step from de sire”.



(a) Reference (example 4)



(b) RVQ (example 4)



(c) SCR (example 4)

Figure 8: **Mel-spectrogram comparison for example 4.** Reference recording and speech synthesized from RVQ and sparse continuous representations (SCRs) for “and shoots the bird which sat is fies his hunger in mod ern con di tions the re la tions are vast ly more com pli cat ed”.