BRIDLE: Generalized Self-supervised Learning with Quantization

Anonymous authors
Paper under double-blind review

Abstract

Self-supervised learning (SSL) has been a powerful approach for learning meaningful representations from unlabeled data across various domains, reducing the reliance on large labeled datasets. Inspired by BERT's success in capturing deep bidirectional contexts in natural language processing, similar frameworks have been adapted to other modalities such as audio, with models like BEATs extending the bidirectional training paradigm to audio signals using vector quantization (VQ). However, these frameworks face challenges, notably their dependence on a single codebook for quantization, which may not capture the complex, multifaceted nature of signals. In addition, inefficiencies in codebook utilization lead to underutilized code vectors. To address these limitations, we introduce BRIDLE (Bidirectional Residual Quantization Interleaved Discrete Learning Encoder), a self-supervised encoder pretraining framework that incorporates residual quantization (RQ) into the bidirectional training process, and is generalized for pretraining with audio, image, and video. Using multiple hierarchical codebooks, RQ enables fine-grained discretization in the latent space, enhancing representation quality. BRIDLE involves an interleaved training procedure between the encoder and tokenizer. We evaluate BRIDLE on audio understanding tasks using classification benchmarks, achieving state-of-the-art results, and demonstrate competitive performance on image classification and video classification tasks, showing consistent improvements over traditional VQ methods in downstream performance.

1 Introduction

By understanding the intrinsic structures within the data, *Self-supervised learning* (SSL) methods have reduced the reliance on large labeled datasets, which are often costly and time-consuming to produce. One of the most influential models in this space is BERT (Devlin et al., 2019), which has set new benchmarks in natural language processing tasks by capturing deep bidirectional contexts.

In the context of audio representation, capturing the intrinsic details of the audio input can be used in various downstream tasks (Niizumi et al., 2021). The introduction of transformer-based architectures and masked modeling approaches have revolutionized the field, with pioneering models such as SSAST (Gong et al., 2022a), MAE-AST (Baade et al., 2022), Audio-MAE (Huang et al., 2022). These models have demonstrated that discretizing continuous audio signals into discrete tokens can produce high-quality representations that are effective for a range of downstream tasks, including audio classification, speech recognition, and sound event detection (Chen et al., 2023; Baevski et al., 2020b; Hsu et al., 2021). BEATs (Chen et al., 2023) extends the bidirectional training paradigm to audio signals, achieving state-of-the-art results in various audio understanding tasks, using the idea of representing the inputs with *Vector Quantization* (VQ) and training the encoder to predict the mapped tokens in an unsupervised context.

However, despite their strengths, these frameworks face several challenges that limit their potential. Notably, their dependence on a single codebook for quantization may not capture the complex, multi-dimensional nature of audio signals, particularly in environments with diverse sound sources or varying acoustic conditions (van den Oord et al., 2017). Additionally, the training processes for the codebooks can be inefficient, as some code vectors remain unused or underutilized, leading to suboptimal representation learning and increased

training time (Razavi et al., 2019). Several techniques have been introduced to optimize codebook utilization in VQ and *Residual VQ* (RVQ) frameworks (Kumar et al., 2023; Łańcucki et al., 2020), including k-means initialization for codebook vectors, randomized restarts for unused codebooks over several batches, and quantizer dropout to improve codebooks learning and usage.

Residual Quantization (RQ) has emerged as an effective method in various machine learning applications, particularly in the domain of computer vision and audio understanding, where it serves as a robust tool to reduce the dimensionality of data representations. Originally developed in the context of signal processing and vector quantization (Gray & Neuhoff, 1998; Gersho & Gray, 2012), RQ operates by recursively quantizing the residuals of previously quantized vectors, allowing it to capture fine-grained details that are missed by single-step quantization. This method has been successfully applied in large-scale image retrieval (Jegou et al., 2010) and approximate nearest neighbor search (Ge et al., 2013), demonstrating its efficiency in handling high-dimensional data. This hierarchical approach to quantization has proven to be highly beneficial in reducing information loss, making it an attractive option for tasks such as image compression and feature representation in vision models (Razavi et al., 2019; Lee et al., 2022).

ImageNet dataset Russakovsky et al. (2015) has been instrumental in advancing computer vision, serving as a benchmark for image classification tasks. Models pre-trained on ImageNet (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016) have achieved remarkable performance and are widely used for transfer learning in various downstream tasks. Self-supervised learning methods on ImageNet have further improved the quality of learned representations without relying on labeled data. Contrastive learning frameworks such as SimCLR (Chen et al., 2020) and MoCo (He et al., 2020) maximize agreement between differently augmented views of the same image, capturing invariant features. Masked image modeling approaches such as MAE (He et al., 2022) apply masked modeling techniques to images, similar to BERT's masked language modeling.

Interestingly, audio understanding tasks can often be modeled similarly to vision models, as audio signals can be transformed into visual representations like mel-spectrograms. By converting raw audio data into mel-spectrograms, which encode frequency and temporal information as 2D matrices, the task of understanding audio becomes analogous to image processing. Residual quantization, when applied to image representations, helps to capture fine spectral details, much like it captures intricate visual features in image data. Tokenization and quantization methods have been utilized for efficient vector quantization in self-supervised learning frameworks such as those used in speech recognition and audio synthesis (van den Oord et al., 2017; Dhariwal et al., 2020). By quantizing the latent spaces of mel-spectrograms, models can efficiently capture important temporal and spectral patterns, crucial for audio classification and speech tasks (Baevski et al., 2020b; Chen et al., 2023; Chiu et al., 2022). This cross-domain applicability highlights residual quantization's versatility and its critical role in both vision and audio-based models, where it ensures efficient representation learning and high-quality reconstructions.

For video understanding, the Kinetics dataset (Kay et al., 2017) provides a large-scale collection of annotated video clips. Architectures employing 3D convolutions have benefited from pretraining on Kinetics, advancing action recognition and video analysis. Self-supervised approaches (Sun et al., 2019) and contrastive video representation learning methods (Qian et al., 2021) have extended bidirectional and self-supervised learning techniques to the temporal dimension, capturing complex spatiotemporal patterns.

Motivated by the quantization techniques in computer vision and audio compression, we introduce the Bidirectional Residual Quantization Interleaved Discrete Learning Encoder, or BRIDLE for short, which is built and improved based on BEATs (Chen et al., 2023), a bidirectional training process involves self-distilled training where the encoder acts as the teacher for the tokenizer, and the tokenizer trains the encoder to predict the tokens. In BRIDLE, we introduce a joint training framework and some improvements for codebooks usage and training. Our contributions are:

1. We incorporate residual quantization into BEATs framework (Chen et al., 2023) to enhance representation quality by utilizing multiple codebooks in a hierarchical manner, enabling a finer discretization of the audio latent space. Hence, we introduce BRIDLE, a self-supervised training framework with an interleaved training procedure between the main encoder and the tokenizer.

2. We provide comprehensive evaluations of the framework in audio understanding through classification tasks on popular benchmarks, AudioSet (Gemmeke et al., 2017) and ESC-50 (Piczak, 2015), where we demonstrate state-of-the-art results. Additionally, we show competitive performance in image classification tasks with experiments on ImageNet-1K and in video classification with Kinetics-400. We also show consistent improvements in the encoder's downstream performance when using RQ compared to VQ.

3. We present a comprehensive analysis of codebook training within the BRIDLE training framework, covering aspects such as uniform weight initialization vs k-means, initial normalization of input embeddings, and resetting unused codes to prevent stagnation and encourage exploration.

2 Related Work

Self-supervised learning has become a cornerstone in representation learning across various data modalities, including audio, image, and video. In this section, we review the relevant literature in each modality, emphasizing residual quantization techniques, pretraining strategies, and self-supervised learning methods.

In computer vision, masked image modeling approaches, inspired by BERT, have been introduced with models such as BEiT (Bao et al., 2022; Peng et al., 2022; Wang et al., 2023) and MAE (He et al., 2022), where portions of the image are masked, and the model is trained to reconstruct the missing parts. These methods capture both local and global structures in images. Quantization techniques such as VQ-VAE (van den Oord et al., 2017) and its variants have been applied to image data, enabling discrete latent representations that facilitate powerful generative models. Residual quantization further enhances this by using multiple codebooks to capture complex image details, as seen in models such as RQ-VAE (Lee et al., 2022).

Several approaches have been proposed in the context of self-supervised learning for audio representation, with the utilization of contrastive learning, masked prediction, and quantization-based methods. Contrastive and clustering learning methods (Gong et al., 2022b; Chen et al., 2020; van den Oord et al., 2018; Seth et al., 2023) have been adapted to audio to learn discriminative representations (Saeed et al., 2021), which utilize contrastive loss functions to maximize similarity between positive pairs (augmented versions of the same audio clip) while minimizing similarity between negative pairs (different audio clips). These methods employ data augmentations specific to audio, such as time-shifting, pitch shifting, and noise addition, to enhance the robustness of learned features.

Recently, converting audio signals to discrete tokens processed by large language models (LLMs) has gained popularity due to the increasing demand for multimodal understanding. Llama 3 (Dubey et al., 2024) converts audios to tokens and the model could understand both text and audio seamlessly. Given the success of diffusion models on image generations, many works have explored audio generation using diffusions (Kreuk et al., 2023). However, modeling the raw audio waveform is prohibitively expansive for diffusion models, and different compact representations have been explored. AudioLM (Borsos et al., 2023) generates audio samples conditioned on text inputs, operating on discrete learned audio representations. SoundStream (Zeghidour et al., 2021) introduces an end-to-end neural audio codec that encodes audio into discrete tokens suitable for downstream tasks. These approaches facilitate integration of audio data into LLMs, enabling advanced capabilities like audio-based question answering and generation.

Quantization-based techniques, particularly Vector Quantization (VQ), have been central to recent advances in self-supervised audio models. VQ techniques, as used in models such as VQ-Wav2Vec, map high-dimensional audio data into a finite set of discrete tokens or codes, facilitating robust speech representation learning (Baevski et al., 2020a). A larger codebook is needed to capture finer details of the audio signal. Nevertheless, scaling up code book in VQ suffers many challenges, especially skewed of codebook usage with a larger codebook. Instead, RQ, an extension of VQ, employs multiple codebooks in a hierarchical manner to capture finer details in audio data, improving the discretization of the latent space (Dhariwal et al., 2020).

On the other hand, video understanding poses unique challenges due to the additional temporal dimension. Pretraining frameworks have significantly advanced video representation learning through both supervised and self-supervised methods. Several models have been introduced to capture 3D information within videos, such as C3D (Tran et al., 2015), which uses 3D convolutions to learn spatiotemporal features. The two-stream architecture (Simonyan & Zisserman, 2014) processes spatial and temporal information separately using RGB

frames and optical flow. I3D Carreira & Zisserman (2017) inflates 2D convolutional filters pre-trained on ImageNet into 3D, effectively transferring knowledge from images to videos.

Self-supervised learning has been adapted to video to leverage unlabeled data. Temporal order prediction methods such as Shuffle and Learn (Misra et al., 2016) and OPN (Lee et al., 2017) learn representations by predicting the correct temporal order of shuffled frames. Contrastive learning frameworks such as CVRL (Qian et al., 2021) extend contrastive methods to video by treating clips from the same video as positives. Masked video modeling approaches, such as VideoMAE (Tong et al., 2022), mask portions of the video input and train the model to reconstruct them, capturing spatiotemporal dependencies. While less explored in video, residual quantization techniques have potential for efficient video representation. Incorporating RVQ can improve the discretization of spatiotemporal data, facilitating tasks such as video compression and generative modeling.

BRIDLE represents a self-supervised learning framework that can work for all modalities, employing an encoder trained by a bidirectional pretraining process in which the encoder and tokenizer train each other in a self-distilled manner. BRIDLE leverages residual quantization to discretize continuous signals from image, audio and video data, and predicts tokens, akin to masked language models in NLP.

3 BRIDLE framework

The proposed BRIDLE framework focuses on integrating residual quantization Lee et al. (2022), improving codebook representation capability. BRIDLE comprises four main components:

Main Encoder $E(\cdot;\theta_E)$: Maps input audio features $\mathbf{x} \in \mathbb{R}^{T\times F}$, where T is the time dimension and F is the feature dimension, to a latent representation $\mathbf{z} = E(\mathbf{x};\theta_E) \in \mathbb{R}^{T\times D}$, where D is the dimension of the latent space.

Tokenizer $T(\cdot;\theta_T)$: Contains a tokenizer encoder and a set of codebooks, which discretize the latent representation $\mathbf{z}$ into a sequence of discrete tokens $\mathbf{q} = T(\mathbf{z};\theta_T) \in \mathbb{Z}^{T\times M}$, where M is the number of codebooks.

Main Decoder $D(\cdot;\theta_D)$: Predicts the tokens output by the tokenizer, facilitating the reconstruction of the input.

Tokenizer Estimator $\mathrm{TE}(\cdot;\theta_{\mathrm{TE}})$: Predicts the encoder's embeddings from the tokenizer's outputs, ensuring alignment between the encoder and tokenizer.

We adopt the interleaved training framework from BEATs (Chen et al., 2023), which consists of the encoder training phase, and the tokenizer training phase. When we train the encoder, the main encoder and decoder learn to predict masked labels produced by the tokenizer. On the other hand, when the tokenizer is trained, the tokenizer encoder, codebooks, and the estimator learn the embeddings produced by their teacher, the main encoder. Figure 1 details the framework training process. The framework begins with the encoder training phase, utilizing a cold start tokenizer for the first iteration. In subsequent iterations, the tokenizer is reset and trained during a dedicated tokenizer training phase, each followed by an encoder training phase. This approach establishes an interleaved training framework. We evaluate the main encoders in downstream tasks after every encoder training phase for all modes.

3.1 Residual Quantization

To enhance the discretization process, we employ residual quantization in BRIDLE. RQ uses a hierarchy of codebooks $\{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_M\}$, where each codebook $\mathcal{C}_m \in \mathbb{R}^{K_m\times D}$ contains K_m code vectors of dimension D. The quantization process is performed in multiple stages, such that every codebook quantizes the residual error from the previous stage (Razavi et al., 2019).

Given a latent vector $\mathbf{z}_t \in \mathbb{R}^D$ at time step t, the quantization at each stage m is defined as follows. Initially, the residual is set to be $\mathbf{e}_1 = \mathbf{z}_t$. At each stage m, we select the code vector $\mathbf{c}^{(m)}_{i^*(m)} \in \mathcal{C}_m$ that minimizes the

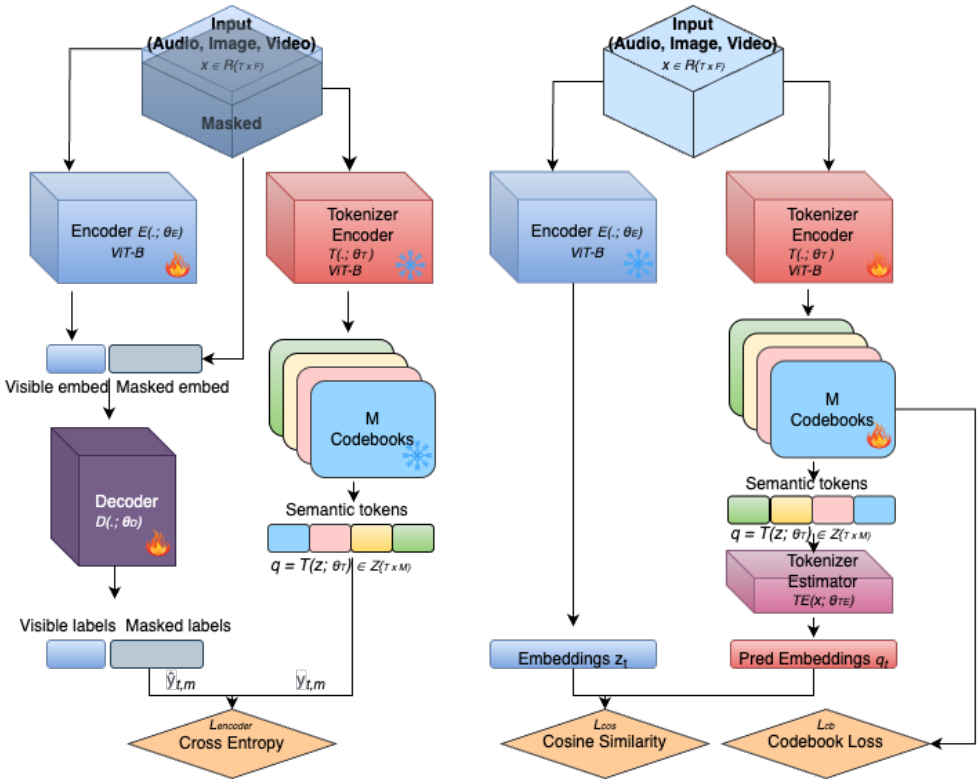


Figure 1: BRIDLE training framework, which contains an encoder training phase (left), and a tokenizer training phase (right)

distance to the residual error $\mathbf{e}_m$, for $\mathbf{c}_i^{(m)} \in \mathcal{C}_m$:

$$i^*(m) = \arg \min_{i=1,2,\ldots,K_m} \left\| \mathbf{e}_m - \mathbf{c}_i^{(m)} \right\|^2. \tag{1}$$

We then update the residual for the next stage:

$$\mathbf{e}_{m+1} = \mathbf{e}_m - \mathbf{c}_{i^*(m)}^{(m)}. \tag{2}$$

After M stages, the final quantized representation $\mathbf{q}_t$ is obtained by summing the selected code vectors from all stages:

$$\mathbf{q}_t = \sum_{m=1}^{M} \mathbf{c}_{i^*(m)}^{(m)}. \tag{3}$$

3.2 Loss Functions

The training of the encoder and tokenizer involves two different set of loss functions. The *encoder loss* is a masked cross-entropy loss that measures the discrepancy between the tokens predicted by the main decoder and the actual tokens output by the tokenizer. This loss ensures that the encoder learns to produce representations that the decoder can accurately translate back into the original token sequence:

$$\mathcal{L}_{\text{encoder}} = -\frac{1}{T} \sum_{t=1}^{T} \sum_{m=1}^{M} \mathbf{y}_{t,m} \log \hat{\mathbf{y}}_{t,m}, \tag{4}$$

where $\mathbf{y}_{t,m}$ is the ground truth token from the tokenizer at time step t and codebook m, $\hat{\mathbf{y}}_{t,m}$ is the predicted probability distribution over the token vocabulary by the main decoder at time step t and codebook m. The *tokenizer loss* comprises two components:

Codebook Loss ($\mathcal{L}_{\text{cb}}$): Encourages effective and accurate mapping and representation of the input vectors by penalizing the distance between the encoder's latent vectors and their corresponding representation chosen from the codebooks. It includes a commitment term to ensure that the encoder output stays close to the code vectors:

$$\mathcal{L}_{\text{cb}} = \frac{1}{T}\sum_{t=1}^{T}\left(\|\text{sg}[\mathbf{z}_t] - \mathbf{q}_t\|^2 + \beta\,\|\mathbf{z}_t - \text{sg}[\mathbf{q}_t]\|^2\right), \tag{5}$$

where $\text{sg}[\cdot]$ denotes the stop-gradient operator, which is identity at forward pass and has zero partial derivatives (van den Oord et al., 2017), and β is a hyperparameter controlling the strength of the commitment loss.

Cosine Similarity Loss ($\mathcal{L}_{\text{cos}}$): Measures the cosine similarity between the tokenizer estimator's output and the main encoder's embeddings, promoting alignment between the two components:

$$\mathcal{L}_{\text{cos}} = 1 - \frac{\sum_{t=1}^{T}\text{TE}(\mathbf{q}_t)\cdot\mathbf{z}_t}{\sum_{t=1}^{T}\|\text{TE}(\mathbf{q}_t)\|\|\mathbf{z}_t\|}. \tag{6}$$

This loss component ensures the tokenizer captures proper latent information given by its teacher, the main encoder, through the training process of predicting the target embeddings. The total tokenizer loss is the sum of the codebook loss (5) and the cosine similarity loss (6):

$$\mathcal{L}_{\text{tokenizer}} = \mathcal{L}_{\text{cb}} + \lambda_{\text{cos}}\mathcal{L}_{\text{cos}}, \tag{7}$$

where λ_{cos} is a weighting factor balancing the two loss components.

3.3 Codebook Training Techniques

To optimize codebook training, we adopt the following key techniques within the BRIDLE framework:

Exponential Moving Average (EMA) update (van den Oord et al., 2017): The EMA update has been shown to be effective in codebook learning (Chen et al., 2023; Lee et al., 2022). The EMA provides smoother updates and reduces variance, leading to more stable convergence. In Appendix D, we show the convergence of the EMA update for all potential codebooks, given minor assumptions on the convergence of latent vectors and sufficiently large number of iterations. To our knowledge, this is a novel theoretical contribution.

Resetting Unused Codes (Kumar et al., 2023): To improve codebook mapping, we reset unused or underutilized codes in the codebook $\mathcal{C}_m$. If the usage count N_i for a code vector $\mathbf{c}_i$ falls below a threshold T_u, we reinitialize $\mathbf{c}_i$ based on the current data distribution:

$$\text{Rand}\{\mathbf{z}_t\} \to \mathbf{c}_i \quad \text{if} \quad N_i < T_u,$$

where $\{\mathbf{z}_t\}$ is the input latent vectors. This prevents codebook collapse and encourages diverse representation learning. In our implementation, resetting unused codes is applied only in the first batch of the tokenizer pretraining phase, so that it does not interfere with codebook training.

Code Embedding Initialization - Uniform or k-means: At the start of training, we initialize codebook entries either uniformly based on the variance of the input features $\{\mathbf{z}_t\}$, or by k-means algorithm on the first batch. Due to the typical usage of normalization for the input latent vectors, the uniform initialization strategy uses the distribution of the range $[-1, 1]$. By sampling from a distribution that matches the input characteristics, we ensure balanced usage of the codebook entries and avoid dominance by a few entries. This ensures that the initial code embeddings are well-bounded.

4 Experiments and Results

We evaluate the BRIDLE model across three modalities: audio, image, and video. For each modality, we perform pretraining on large-scale datasets and fine-tuning on standard benchmarks to assess the model's performance comprehensively. Our goal is to demonstrate that integrating residual quantization, improving codebook training leads to performance improvement in self-supervised representation learning. For important

audio benchmark on AudioSet (Gemmeke et al., 2017), and image benchmark on ImageNet-1K (Russakovsky et al., 2015), we also provide linear probing performances. These results demonstrate more noticeable improvements, especially where the model's fine-tuning results are relatively saturated, given that we are using the base ViT architecture for all modes. For audio, we experiment our framework on the *AudioSet* dataset (Gemmeke et al., 2017), which contains over 2 million 10-second sound clips drawn from YouTube videos and labeled across 527 classes, and the *ESC-50* dataset (Piczak, 2015), a collection of 2,000 5-second environmental audio recordings across 50 classes. For image data, we use the popular *ImageNet-1K* dataset (Deng et al., 2009), which consists of approximately 1.28 million training images and 50,000 validation images in 1,000 classes. For video data, we employ the *Kinetics-400* dataset (Kay et al., 2017), which contains around 240,000 video clips annotated with 400 human action classes.

4.1 Model Architecture

We use the base ViT architectures for all modes. The tokenizer encoder has the same architecture as the main encoder. Similar to BEATs (Chen et al., 2023) and inspired by the success of random linear projection tokenizer in the literature (Chiu et al., 2022; Dubey et al., 2024), for iteration 1, we use a simple linear projection as the cold start tokenizer encoder. The BRIDLE framework incorporates residual quantization with $M = 4$ codebooks for each modality. We test this architecture against the VQ version of the model, where we use $M = 1$ codebook, while maintaining the same total number of codes for fair comparison. For RQ, each latent vector is mapped to $M = 4$ codes, which yield richer representations. We refer to Appendix A for hyperparameters details, Appendix C to support hyperparameters choices and Appendix E for model configurations.

4.2 Training Procedure

We first pretrain the models for two bidirectional iterations, then evaluate the models on corresponding classification tasks on the datasets mentioned above. Due to the empirical observations from BEATs, the performance of the model shows negligible improvements with additional training iterations. We observe similar trend in our experiments, hence we stop at 2 iterations for all modes. During the encoder training phase of audio and image, we use masking ratio of 0.8, and for the tokenizer training phase, we do not apply masking. For the video modality, the process is similar to audio and image training, except for the usage of tube masking with 0.9 ratio to increase reconstruction difficulty and prevent information leakage (Tong et al., 2022).

Audio: The model is pretrained on the AudioSet-2M dataset for 130 epochs per encoder training phase, and 30 epochs for tokenizer training. We then evaluate the model with fine-tuning and linear probing performances on AudioSet-2M and AudioSet-20K, as well as fine-tuning performance on ESC-50.

Image: The model is pretrained on ImageNet-1K following the procedure in AudioMAE (Huang et al., 2022). We use 400 encoder training epochs and 100 tokenizer training epochs. Standard data augmentation techniques such as random cropping and horizontal flipping are applied. After pretraining, we fine-tune the model on the ImageNet-1K training set and evaluate it on the validation set.

Video: We pretrain the model on the Kinetics-400 training set for 800 epochs per encoder training phase, and 200 epochs for tokenizer training phase, following the setup from VideoMAE (Tong et al., 2022). During training, video clips are sampled with temporal strides to capture motion dynamics effectively. Data augmentations include random cropping and horizontal flipping in both spatial and temporal dimensions. Hence, our video models behave like multi-frame image models. After pretraining, the model is fine-tuned on the Kinetics-400 training set and evaluated on the validation set.

4.3 Results

We evaluate the model using mean Average Precision (mAP) for multi-label classification tasks on AudioSet, and classification accuracy with 5-fold cross validation for ESC-50, which is a multi-class single-label dataset. We report Top-1 and Top-5 classification accuracies on the ImageNet-1K validation set. For Kinetics-400, we evaluate the model using Top-1 and Top-5 classification accuracies on the validation set. We present the

audio classification results on the AS-2M, AS-20K, and ESC-50 datasets in Table 1. We compare different models using both fine-tuning and linear probing evaluations. Table 1 demonstrates that incorporating *residual quantization* (RQ) into the BEATs framework enhances performance across all evaluated datasets and metrics. The reproduction of BEATs (Chen et al., 2023) on ∼15% missing data from AudioSet-2M shows a minor drop of ∼0.5% in fine-tuning mAP. Nevertheless, BRIDLE utilizing RQ codebooks and k-means initialization achieves the best results across all evaluation tasks. We have shown improvements compared to the benchmark of BEATs reproduction, which provided state-of-the-art results in the context of audio self-supervised training.

Table 1: Audio classification results on the AS-2M, AS-20K, and ESC-50 datasets. Fine-tuning (FT) and linear probing (LP) results are reported in mAP or accuracy (%).

Model	AS-2M mAP		AS-20K mAP		ESC-50 FT
	FT	LP	FT	LP	mAP | Acc
(VQ) BEATs$_{\text{iter1}}$ Chen et al. (2023)	46.60	26.76	35.36	23.32	97.19 | 93.75
(VQ) BEATs$_{\text{iter2}}$ Chen et al. (2023)	47.64	27.04	36.89	23.41	97.86 | 94.55
(uniform) BRIDLE$_{\text{iter1}}$	47.19	29.15	36.99	25.54	97.47 | 94.45
(uniform) BRIDLE$_{\text{iter2}}$	47.56	28.71	37.72	25.51	97.83 | 94.85
(k-means) BRIDLE$_{\text{iter1}}$	47.17	**29.58**	37.03	26.07	97.61 | 94.45
(k-means) BRIDLE$_{\text{iter2}}$	**47.99**	29.56	**38.08**	**26.37**	**97.89** | **94.95**

In the vision context, the image classification results on the ImageNet-1K dataset are presented in Table 2, where we report both fine-tuning and linear probing accuracies. The fine-tuning action recognition results on the Kinetics-400 dataset are presented in Table 3. We report fine-tuning Top-1 and Top-5 accuracies.

Table 2: Image classification results on the ImageNet-1K dataset. Fine-tuning and linear probing results are reported in Top-1 and Top-5 accuracies (%).

Model	FT Acc (%)		LP Acc (%)	
	Top-1	Top-5	Top-1	Top-5
(VQ) BRIDLE$_{\text{iter1}}$	79.10	93.86	44.82	69.22
(VQ) BRIDLE$_{\text{iter2}}$	80.26	94.41	53.21	76.89
(k-means) BRIDLE$_{\text{iter1}}$	80.00	94.30	52.96	76.60
(k-means) BRIDLE$_{\text{iter2}}$	**81.10**	**95.00**	**56.30**	**79.24**

Table 3: Action recognition results on the Kinetics-400 dataset. Fine-tuning results are reported in Top-1 and Top-5 accuracies (%).

Model	Top-1 Acc (%)	Top-5 Acc (%)
(VQ) BRIDLE$_{\text{iter1}}$	68.69	87.44
(VQ) BRIDLE$_{\text{iter2}}$	71.32	88.86
(k-means) BRIDLE$_{\text{iter1}}$	71.36	89.03
(k-means) BRIDLE$_{\text{iter2}}$	**72.90**	**90.08**

Table 2 and Table 3 highlight the effectiveness of residual quantization in enhancing image and video representation learning, leading to better performance in action recognition tasks. Across all three modalities—audio, image, and video, the incorporation of residual quantization into the bidirectional pretraining framework consistently improves performance. The benefits are more pronounced in linear probing evaluations, suggesting that RQ enables the encoder to learn more generalizable and robust representations. The use of k-means initialization for codebooks further enhances performance, indicating that careful codebook initialization is crucial for optimizing quantization-based models.

5 Empirical Discussions

Codebook enhancements: A major component of the framework is the codebook, for which we need to adjust the size and number of codebooks. Hence, a natural question is whether we can further improve the model's performance by enhancing codebooks' size, or increasing the number of codes for each input latent vector. This can potentially introduce sparsities in the efficiency of the quantization process and can degrade model performance. We have observed performance degradation in audio and image context if we increase the number of codes, or use *soft codes*, i.e. assigning each input embedding to the top-k closest code vectors based on distance metrics. These results suggest that while large codebooks and soft codes can provide a powerful tokenization process, they may not provide performance benefits in our framework. This outcome may result from the dilution of meaningful features across too many code vectors, or the limitation in the model's capability to learn a large space of tokens, given an encoding embedding of dimension 768 from our base ViT architecture. Additionally, existing codebook methods assume hard-assigned code, and uniform weighting method is shown ineffective. A weighted combination of a sparse subset of codes could be a solution worth exploring in future works (Aharon et al., 2006). Therefore, determining an optimal codebook size is crucial.

Is joint training of the encoder and the tokenizer possible? A disadvantage of the BRIDLE pretraining framework is the long training process, where we interchangeably train the encoder E and tokenizer T. To accelerate training, we consider a joint training strategy within BRIDLE, where the encoder E and tokenizer T are updated simultaneously within each iteration (Grill et al., 2020). This process allows the model components to co-adapt, fostering faster convergence. The loss functions of the two training phases can be combined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{encoder}} + \alpha \mathcal{L}_{\text{tokenizer}}, \tag{8}$$

where α is a loss ratio between the encoder training and tokenizer training. In BEATs' training framework, the main encoder and the tokenizer are trained in a leaved way: update one while the other is frozen (Chen et al., 2023). Additionally, in every bidirectional training iteration, the tokenizer including its codebooks are reset, with the intuition of improving codebook usage and audio feature capturing. In our empirical results, training all architectures simultaneously provides almost equivalent results. Furthermore, to stabilize codebook training, it is beneficial to update the tokenizer for every few steps of training the main encoder. We have observed that $\alpha = 0.5$ and updating the tokenizer every 5 steps works well for VQ. Nevertheless, we find difficulties in tuning the hyperparameters for RQ models to perform well on downstream tasks, despite the codebooks converge nicely. Hence, we leave this idea for future work.

Best practices for codebook training: Our empirical studies identified several techniques that enhance codebook training within the BRIDLE framework:

1. EMA updates: The EMA update for codebooks demonstrates higher performance than standard backpropagation.

2. k-means initialization: Initializing codebooks using k-means clustering significantly outperforms uniform initialization. This method ensures that code vectors are better aligned with the data distribution from the outset.

3. Input embedding normalization: Normalizing input embeddings at the beginning of every codebook layer helps stabilize training and improves quantization quality by maintaining consistent scaling across embeddings.

Codebook evaluations: To validate the effectiveness of codebook's representations, we evaluate the codebooks on *Code Usage Rate* (CUR), and *Effective Code Usage* (ECU) (See Appendix G for formulations). These metrics help to systematically assess the diversity and balance of code usage across different configurations. We observe that RQ has superior CUR of $\sim$100%, which intuitively can be easier to achieve with smaller size codebooks compared to VQ. It can be additionally observed that k-means initialization can generally yields $\sim$100% CUR without training the codebooks. Furthermore, in the ECU metric, RQ shows better performance than VQ, demonstrating better code utilization.

6 Conclusion and Future Work

In this paper, we introduced *BRIDLE*, a self-supervised pretraining framework that incorporates residual quantization into the bidirectional encoder paradigm. Using multiple codebooks hierarchically, our approach enables finer discretization of the latent space, enhancing representation quality across audio, image, and video modalities.

Our experiments demonstrated that BRIDLE consistently outperforms traditional vector quantization methods in both fine-tuning and linear probing evaluations. Specifically, we achieved state-of-the-art results on audio classification benchmarks on AudioSet, and showed competitive results on the ImageNet-1K and Kinetics-400 datasets for image and video classification tasks, respectively. The incorporation of residual quantization not only improved performance but also enhanced the generalization ability of the encoder representations.

We conducted a comprehensive analysis of codebook training within the BRIDLE framework, identifying effective strategies such as k-means initialization, input embedding normalization, and the use of exponential moving average updates for codebook vectors. These techniques contributed to better codebook utilization and stability during training. By addressing these directions, we aim to further enhance the capabilities of self-supervised learning frameworks and contribute to the development of more robust and efficient models across diverse modalities. While BRIDLE shows promising results, there are several avenues for future exploration, such as joint training framework for efficiency, enhanced codebook learning, or apply the framework in cross-modalities settings.

References

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

Alan Baade, Puyuan Peng, and David Harwath. MAE-AST: Masked autoencoding audio spectrogram transformer. In *Proceedings of Interspeech 2022*, pp. 2438–2442, 2022.

Alexei Baevski, Steffen Schneider, and Michael Auli. vq-wav2vec: Self-supervised learning of discrete speech representations. In *International Conference on Learning Representations (ICLR)*, 2020a.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12449–12460. Curran Associates, Inc., 2020b.

Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. BEiT: BERT pre-training of image transformers. In *International Conference on Learning Representations*, 2022.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, and Neil Zeghidour. AudioLM: a language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533, 2023.

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.

Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 6299–6308, 2017.

Sanyuan Chen, Yu Wu, Chengyi Wang, Shujie Liu, Daniel Tompkins, Zhuo Chen, Wanxiang Che, Xiangzhan Yu, and Furu Wei. BEATs: audio pre-training with acoustic tokenizers. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 5178–5193. PMLR, 2023.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 1597–1607. PMLR, 2020.

Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9640–9649, 2021.

Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu. Self-supervised learning with random-projection quantizer for speech recognition. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162, pp. 3915–3924. PMLR, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE, 2009.

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

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Tiezheng Ge, Kaiming He, Qifa Ke, and Jian Sun. Optimized product quantization for approximate nearest neighbor search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2946–2953, 2013.

Jort F Gemmeke, Daniel PW Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 776–780. IEEE, 2017.

Allen Gersho and Robert M Gray. *Vector Quantization and Signal Compression*, volume 159. Springer Science & Business Media, 2012.

Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. SSAST: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 10699–10709, 2022a.

Yuan Gong, Andrew Rouditchenko, Alexander H Liu, David Harwath, Leonid Karlinsky, Hilde Kuehne, and James R Glass. Contrastive audio-visual masked autoencoder. In *The Eleventh International Conference on Learning Representations*, 2022b.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44(6): 2325–2383, 1998.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent-a new approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

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

Po-Yao Huang, Hu Xu, Juncheng Li, Alexei Baevski, Michael Auli, Wojciech Galuba, Florian Metze, and Christoph Feichtenhofer. Masked autoencoders that listen. In *Advances in Neural Information Processing Systems*, volume 35, pp. 28708–28720. Curran Associates, Inc., 2022.

Herve Jegou, Matthijs Douze, and Cordelia Schmid. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):117–128, 2010.

Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi. AudioGen: Textually guided audio generation. In *The Eleventh International Conference on Learning Representations*, 2023.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

Rithesh Kumar, Prem Seetharaman, Alejandro Luebs, Ishaan Kumar, and Kundan Kumar. High-fidelity audio compression with improved RVQGAN. In *Advances in Neural Information Processing Systems*, volume 36, pp. 27980 – 27993. Curran Associates, Inc., 2023.

Adrian Łańcucki, Jan Chorowski, Guillaume Sanchez, Ricard Marxer, Nanxin Chen, Hans JGA Dolfing, Sameer Khurana, Tanel Alumäe, and Antoine Laurent. Robust training of vector quantized bottleneck models. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7. IEEE, 2020.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11523–11532, 2022.

Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 667–676, 2017.

Ishan Misra, C Lawrence Zitnick, and Martial Hebert. Shuffle and learn: unsupervised learning using temporal order verification. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 527–544. Springer, 2016.

Daisuke Niizumi, Daiki Takeuchi, Yasunori Ohishi, Noboru Harada, and Kunio Kashino. Byol for audio: Self-supervised learning for general-purpose audio representation. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. BEiT v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

Karol J Piczak. ESC: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–1018, 2015.

Rui Qian, Tianjian Meng, Boqing Gong, Ming-Hsuan Yang, Huisheng Wang, Serge Belongie, and Yin Cui. Spatiotemporal contrastive video representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6964–6974, 2021.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115:211–252, 2015.

Aaqib Saeed, David Grangier, and Neil Zeghidour. Contrastive learning of general-purpose audio representations. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3875–3879. IEEE, 2021.

Ashish Seth, Sreyan Ghosh, Srinivasan Umesh, and Dinesh Manocha. Slicer: Learning universal audio representations using low-resource self-supervised pre-training. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE, 2023.

Claude Elwood Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27 (3):379–423, 1948.

Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. VideoBERT: A joint model for video and language representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7464–7473, 2019.

Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 10078–10093. Curran Associates, Inc., 2022.

Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4489–4497, 2015.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

Wenhui Wang, Hangbo Bao, Li Dong, Johan Bjorck, Zhiliang Peng, Qiang Liu, Kriti Aggarwal, Owais Khan Mohammed, Saksham Singhal, Subhojit Som, and Furu Wei. Image as a foreign language: BEIT pretraining for vision and vision-language tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19175–19186, 2023.

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, 2021.

BRIDLE: Generalized Self-supervised Learning with Quantization

Appendix

The Appendix is organized as follows.

- In Appendix A, we provide the details of the hyperparameters.
- In Appendix B, we show details of the datasets we use for each mode.
- In Appendix C, we provide the ablation and sensitivity studies to support hyperparameter choices.
- In Appendix D, we present the details of the EMA update for the codebooks.
- In Appendix E, we discuss the model configurations.
- In Appendix F, we discuss the framework performance in comparison with existing self-supervised learning frameworks on ImageNet.
- In Appendix G, we provide the definitions of some metrics that are used in codebook and tokenizer evaluation.

A Hyperparameter Details

Table 4 presents the details of the hyperparameters used in the training of our BRIDLE model.

B Datasets

In this section, we show details of the datasets we conduct the experiments for each mode.

Audio: We use the *AudioSet* dataset (Gemmeke et al., 2017), which contains over 2 million 10-second sound clips drawn from YouTube videos and labeled across 527 classes. For AudioSet, we collect the data from 2023 source, for which due to changes in YouTube content availability, approximately 10–15% of the data is no longer accessible, affecting both the training and evaluation sets. This data loss may introduce variations in performance metrics. Additionally, we use the *ESC-50* dataset (Piczak, 2015), a collection of 2,000 5-second environmental audio recordings across 50 classes, to evaluate the model's generalization ability in environmental sound classification.

Image: For image data, we utilize the *ImageNet-1K* dataset (Deng et al., 2009), which consists of approximately 1.28 million training images and 50,000 validation images in 1,000 classes. ImageNet-1K is a standard benchmark for image classification tasks and enables us to evaluate the model's performance in visual recognition.

Video: For video data, we employ the *Kinetics-400* dataset (Kay et al., 2017), which contains around 240,000 video clips annotated with 400 human action classes. Each clip is approximately 10 seconds long and is sourced from YouTube videos. This dataset allows us to assess the model's capability in capturing spatiotemporal information for action recognition tasks. For our setup, the video frames have dimensions 512 (W) $\times$ 288 (H).

C Ablation and Sensitivity Studies

In the following tables, we provide sensitivity studies with variations of training strategies we tested to arrive at our optimal choices stated in Appendix A. For each table, we use our best results for audio or image experiments at a specific iteration as the benchmark, then state the metric changes as the training strategies vary. "—" represents the same hyperparameter or strategy as the benchmark.

Table 4: Hyperparameters for the BRIDLE Model across different modalities

Hyperparameter	Audio			Image	Video
	AS-2M	AS-20K	ESC-50	ImageNet-1K	Kinetics-400
Pretrain Batch Size (B)		16		16	32
FT & LP Batch Size (B)		8		32	32
Encoder Pretrain LR		5×10^{-4}		5×10^{-4}	1.2×10^{-3}
Tokenizer Pretrain LR		2×10^{-4}		2×10^{-4}	2.5×10^{-4}
FT LR	5×10^{-4}	4×10^{-4}	2×10^{-3}	5×10^{-4}	7.5×10^{-4}
LP LR	4×10^{-3}	4×10^{-2}	—	2×10^{-3}	—
Encoder Pretrain Epochs		130		400	800
Tokenizer Pretrain Epochs		30		100	200
FT Epochs	120	150	300	200	75
LP Epochs	250	400	—	300	—
Pretrain Mask Ratio		0.8		0.8	0.9
FT Masking		2D, ratio 0.2		1D, ratio 0.2	0.0
LP Masking	0.0	0.0	—	0.0	—
Optimizer			AdamW		
Weight Decay			1×10^{-4}		
FT Loss		BCE		BCE	CE
LP Loss	BCE	BCE	—	BCE	—
Normalization Mean	−4.4446096	−4.4446096	−6.6268077	ImageNet default mean	
Normalization Std	3.3216383	3.3216383	5.358466	ImageNet default std	
Codebook Num (M), VQ			1		
Codebook Size (K_m), VQ			1024		
Codebook Num (M), RQ			4		
Codebook Size (K_m), RQ			256		
Codebook Dim (d_m)			256		
EMA Decay Rate (γ)			0.99		
Pretrain nnodes			8		
FT & LP nnodes	4	4	1	4	8
GPU per Node			8		
Code Reset Threshold (T_u)			1		

Table 5: Data availability in AudioSet segments

Data Segment	Original	Obtained	Percentage
Balanced train	22,160	18,685	84.3%
Unbalanced train	2,041,789	1,738,788	85.2%
Evaluation	20,371	17,142	84.1%

Table 6: BRIDLE sensitivity experiments on AudioSet-2M at iteration 1

Codebooks params	Codebook init	normalize codebooks	soft code	imagenet pretrained	FT mAP % chg	LP mAP % chg
4 sets, 256 codes	kmeans	yes	no	no	0 (47.17)	0 (29.58)
—	—	—	4	—	−0.6	−2.37
3 sets, 256 codes	—	no	—	—	−0.4	−0.86
—	—	no	—	—	−0.11	−1.01
—	uniform	—	—	—	+0.02	−0.43
—	uniform	—	no	—	−0.03	−3.66
1 set, 1024 codes	uniform	—	no	yes	−0.33	−9.24

Table 7: BRIDLE sensitivity experiments on AudioSet-2M at iteration 2

Codebooks params	Codebook init	normalize codebooks	codebook training strategy	FT mAP % chg	LP mAP % chg
4 sets, 256 codes	kmeans	yes	EMA	0 (47.99)	0 (29.56)
—	—	no	—	−0.54	−2.48
—	uniform	—	—	−0.43	−0.85
—	uniform	no	—	−0.41	−1.93
—	uniform	no	back prop	−0.65	−3.05
1 set, 1024 codes	uniform	no	—	−0.35	−2.52

D The EMA Update for the Codebooks

To maintain stable updates of the codebook vectors, we employ the EMA strategy (van den Oord et al., 2017). For each code vector $\mathbf{c}_i$ in a codebook $\mathcal{C}_m$, the update rule at every time step t is defined as: [1]

$$n_{i,t} := \sum_{j=1}^{S} \delta_{q_{j,t},i}, \qquad \ell_{i,t} := \sum_{j=1}^{S} \delta_{q_{j,t},i} \cdot \mathbf{z}_{j,t}, \tag{9}$$

$$N_{i,t+1} \leftarrow \gamma N_{i,t} + (1-\gamma) n_{i,t}, \tag{10}$$

$$\hat{N}_{i,t+1} \leftarrow (N_{i,t+1} + \epsilon) \frac{\sum_{i=1}^{K_m} N_{i,t+1}}{\sum_{i=1}^{K_m} N_{i,t+1} + K_m \cdot \epsilon}, \tag{11}$$

$$\mathbf{m}_{i,t+1} \leftarrow \gamma \mathbf{m}_{i,t} + (1-\gamma) \ell_{i,t}, \tag{12}$$

$$\mathbf{c}_{i,t+1} \leftarrow \frac{\mathbf{m}_{i,t+1}}{\hat{N}_{i,t+1}}, \tag{13}$$

where:

- $\gamma \in (0, 1)$ is the EMA decay rate.
- S is the total number of samples (sample size) at every time step.
- $\delta_{q_{j,t},i}$ is the Kronecker delta function, which equals to 1 if $q_{j,t} = i$ and 0 otherwise, where $q_{j,t}$ represents the code that is mapped to the latent vector $\mathbf{z}_{j,t}$.
- $\mathbf{z}_{j,t}$ is the latent vector of sample j at time t.
- $N_{i,t}$ is the accumulated usage count (cluster size) of the code vector $\mathbf{c}_i$, where $N_{i,0} = 0$.
- $\mathbf{m}_{i,t}$ is the running embedding update for code vector $\mathbf{c}_{i,t}$, where $\mathbf{m}_{i,0} = \mathbf{c}_{i,0} = \mathbf{c}_0$ for some vector $\mathbf{c}_0$ of dimension d_m.

[1] Note that $\mathbf{c}_i$ depends on m. To simplify the notation, we write $\mathbf{c}_i$ instead of $\mathbf{c}_i^{(m)}$ to hide the dependence on m. Similarly, for the update rule for $n_{i,t}, N_{i,t}, \hat{N}_{i,t}, \mathbf{m}_{i,t}, \mathbf{c}_{i,t}$ in (9)-(13), we hide the dependence on m to ease the notation.

Table 8: BRIDLE sensitivity experiments on ImageNet-1K at iteration 1

Codebooks params	Codebook init	normalize codebooks	loss function	FT Acc Top-1 % chg	FT Acc Top-5 % chg
4 sets, 256 codes	kmeans	yes	BCE	0 (80.00)	0 (94.30)
—	—	—	CE	−0.97	+0.17
4 sets, 512 codes	—	no	—	−0.21	−0.09

Table 9: BRIDLE sensitivity experiments on ImageNet-1K at iteration 2

Codebooks params	Codebook init	normalize codebooks	loss function	FT Acc Top-1 % chg	FT Acc Top-5 % chg
4 sets, 256 codes	kmeans	yes	BCE	0 (80.00)	0 (94.30)
—	—	—	CE	−1.48	−0.07

The EMA update can also be used to stabilize the learning of the codebook vectors in the Residual Quantization (RQ) process that can work directly with mini-batches. Each set of codebook $\mathcal{C}_m$ for $m \in \{1, 2, \ldots, M\}$ can be updated by the above update rule.

Mathematically, the EMA update (9)-(13) can be re-written as

$$N_{i,t+1} = \gamma N_{i,t} + (1-\gamma) n_{i,t} + \epsilon, \tag{14}$$

$$\mathbf{m}_{i,t+1} = \gamma \mathbf{m}_{i,t} + (1-\gamma)\ell_{i,t}, \tag{15}$$

$$\mathbf{c}_{i,t+1} = \frac{\mathbf{m}_{i,t+1}}{\gamma N_{i,t} + (1-\gamma) n_{i,t} + \epsilon} \cdot \frac{\sum_{i=1}^{K_m} (\gamma N_{i,t} + (1-\gamma) n_{i,t}) + K_m \cdot \epsilon}{\sum_{i=1}^{K_m} (\gamma N_{i,t} + (1-\gamma) n_{i,t})}, \tag{16}$$

for any $t = 0, 1, 2, \ldots$ with $N_{i,0} = 0$, $\mathbf{m}_{i,0} = \mathbf{c}_{i,0} = \mathbf{c}_0$, where $n_{i,t}, \ell_{i,t}$ are given in (9).

Next, we provide convergence analysis for the EMA update (9)-(13). Before we proceed, let us show that $\mathbf{m}_{i,t}$ and $N_{i,t}$ are uniformly bounded in t, which will be used in our convergence analysis. Indeed, we will first show the following technical lemma for $\mathbf{m}_{i,t}$.

Lemma 1. *Assume that for every $j = 1, \ldots, S$, $\sup_t \|\mathbf{z}_{j,t}\| < \infty$ almost surely. Then, for any $i = 1, 2, \ldots, K_m$ and $t = 0, 1, 2, \ldots$,*

$$\|\mathbf{m}_{i,t}\| \leq \mathcal{B}_{\mathbf{m}}, \tag{17}$$

almost surely, where

$$\mathcal{B}_{\mathbf{m}} := \|\mathbf{c}_0\| + \sum_{j=1}^{S} \sup_t \|\mathbf{z}_{j,t}\|. \tag{18}$$

Proof. We use mathematical induction to prove (17). First, when $t = 0$, $\mathbf{m}_{i,0} = \mathbf{c}_0$ so that (17) trivially holds. Next, let us assume that $\|\mathbf{m}_{i,t}\| \leq \mathcal{B}_{\mathbf{m}}$ almost surely for every $i = 1, 2, \ldots, K_m$. Then, from (15), we have

$$\begin{aligned} \|\mathbf{m}_{i,t+1}\| &\leq \gamma \|\mathbf{m}_{i,t}\| + (1-\gamma)\|\ell_{i,t}\| \\ &\leq \gamma \mathcal{B}_{\mathbf{m}} + (1-\gamma) \sum_{j=1}^{S} \sup_t \|\mathbf{z}_{j,t}\| \leq \mathcal{B}_{\mathbf{m}}, \end{aligned} \tag{19}$$

almost surely, where we used the definition of $\mathcal{B}_{\mathbf{m}}$ in (18). This completes the proof. □

Next, let us show that $N_{i,t}$ is uniformly bounded.

Lemma 2. *For any $i = 1, 2, \ldots, K_m$ and $t = 0, 1, 2, \ldots$,*

$$N_{i,t} \leq \mathcal{B}_N, \tag{20}$$

almost surely, where

$$\mathcal{B}_N := S. \tag{21}$$

Proof. First, we recall from (14) that

$$N_{i,t+1} = \gamma N_{i,t} + (1-\gamma) n_{i,t} + \epsilon. \tag{22}$$

We use mathematical induction to prove (20). First, when $t = 0$, $N_{i,0} = 0$ so that (20) trivially holds. Next, let us assume that $N_{i,t} \leq \mathcal{B}_N$ almost surely for every $i = 1, 2, \ldots, K_m$. Then, we have

$$N_{i,t+1} \leq \gamma \mathcal{B}_N + (1-\gamma) S + \epsilon = \mathcal{B}_N, \tag{23}$$

almost surely, where we used the definition of $\mathcal{B}_N$ in (21) and the definition of $n_{i,t}$ in (9) such that $n_{i,t} \leq S$ for any i and t. Hence, we proved (20). □

Now, we are ready to show the convergence of the EMA update (9)-(13). We have the following result.

Proposition 3. *Assume that for every $j = 1, 2, \ldots, S$, $q_{j,t}$ and $\mathbf{z}_{j,t}$ converge almost surely as $t \to \infty$. Then, $(N_{i,t}, \mathbf{m}_{i,t}, \mathbf{c}_{i,t})$ converges almost surely to $(N_{i,\infty}, \mathbf{m}_{i,\infty}, \mathbf{c}_{i,\infty})$ as $t \to \infty$ for every $i = 1, 2, \ldots, K_m$, where*

$$\begin{aligned}
\mathbf{m}_{i,\infty} &= \ell_{i,\infty}, \\
N_{i,\infty} &= n_{i,\infty} + \frac{\epsilon}{1-\gamma}, \\
\mathbf{c}_{i,\infty} &= \frac{\ell_{i,\infty}}{n_{i,\infty} + \frac{\gamma\epsilon}{1-\gamma} + \epsilon} \frac{\sum_{i=1}^{K_m} (n_{i,\infty} + \frac{\gamma\epsilon}{1-\gamma} + \epsilon)}{\sum_{i=1}^{K_m} (n_{i,\infty} + \frac{\gamma\epsilon}{1-\gamma})},
\end{aligned}$$

with $\ell_{i,\infty} = \sum_{j=1}^{S} \delta_{q_{j,\infty},i} \cdot \mathbf{z}_{j,\infty}$ and $n_{i,\infty} = \sum_{j=1}^{S} \delta_{q_{j,\infty},i}$.

Proof. First, for every j, $\mathbf{z}_{j,t}$ converges almost surely so that it is bounded almost surely which satisfies the assumption of Lemma 1. It follows from Lemma 1 that for every $i = 1, \ldots, K_m$, $\mathbf{m}_{i,t}$ is uniformly bounded in t almost surely, by Bolzano-Weierstrass theorem, $\mathbf{m}_{i,t}$ has a convergent subsequence. Suppose that its limit is $\mathbf{m}_{i,\infty}$. If we can show that the limit of any subsequence of $\mathbf{m}_{i,t}$ is the same, which is $\mathbf{m}_{i,\infty}$, then it follows that $\mathbf{m}_{i,t}$ converges to $\mathbf{m}_{i,\infty}$ almost surely as $t \to \infty$.

To show this, note that since for every j, $\mathbf{z}_{j,t}$ and $q_{j,t}$ converge almost surely, we have $\ell_{i,t} \to \ell_{i,\infty}$ almost surely as $t \to \infty$, where $\ell_{i,\infty} = \sum_{j=1}^{S} \delta_{q_{j,\infty},i} \cdot \mathbf{z}_{j,\infty}$. Then it follows from (15) that

$$\mathbf{m}_{i,\infty} = \gamma \mathbf{m}_{i,\infty} + (1-\gamma) \ell_{i,\infty}, \tag{24}$$

which implies that $\mathbf{m}_{i,\infty} = \ell_{i,\infty}$ and thus this limit is unique. Hence, we conclude that $\mathbf{m}_{i,t} \to \mathbf{m}_{i,\infty}$ almost surely as $t \to \infty$.

Next, from Lemma 2, for every $i = 1, \ldots, K_m$, $N_{i,t}$ is uniformly bounded in t almost surely, by Bolzano-Weierstrass theorem, $N_{i,t}$ has a convergent subsequence. Suppose that its limit is $N_{i,\infty}$. If we can show that the limit of any subsequence of $N_{i,t}$ is the same, which is $N_{i,\infty}$, then it follows that $N_{i,t}$ converges almost surely to $N_{i,\infty}$ as $t \to \infty$.

To show this, note that since for every j, $q_{j,t}$ converge almost surely, we have $n_{i,t} \to n_{i,\infty}$ as $t \to \infty$, where $n_{i,\infty} = \sum_{j=1}^{S} \delta_{q_{j,\infty},i}$. We notice that if $N_{i,\infty}$ is any limiting point of a subsequence of $N_{i,t}$, then it follows from (14) that

$$N_{i,\infty} = \gamma N_{i,\infty} + (1-\gamma) n_{i,\infty} + \epsilon, \tag{25}$$

for every $i = 1, 2, \ldots, K_m$, By solving (25) for $N_{i,\infty}$, we obtain

$$N_{i,\infty} = n_{i,\infty} + \frac{\epsilon}{1-\gamma}, \tag{26}$$

for every $i = 1, 2, \ldots, K_m$. Therefore, this limit is unique and hence $N_{i,t}$ converges to $N_{i,\infty} := n_{i,\infty} + \frac{\epsilon}{1-\gamma}$ almost surely as $t \to \infty$ for every $i = 1, 2, \ldots, K_m$.

Finally, we recall from (16) that

$$\mathbf{c}_{i,t} = \frac{\mathbf{m}_{i,t}}{\gamma N_{i,t-1} + (1-\gamma) n_{i,t-1} + \epsilon} \cdot \frac{\sum_{i=1}^{K_m} (\gamma N_{i,t-1} + (1-\gamma) n_{i,t-1}) + K_m \cdot \epsilon}{\sum_{i=1}^{K_m} (\gamma N_{i,t-1} + (1-\gamma) n_{i,t-1})}. \tag{27}$$

Note that since for every j, $q_{j,t}$ converge almost surely, we have $n_{i,t} \to n_{i,\infty}$ as $t \to \infty$, where $n_{i,\infty} = \sum_{j=1}^{S} \delta_{q_{j,\infty},i}$. Moreover, since we already proved that $\mathbf{m}_{i,t}$ converges to $\mathbf{m}_{i,\infty}$ and $N_{i,t}$ converges to $N_{i,\infty}$ almost surely as $t \to \infty$, we conclude that $\mathbf{c}_{i,t}$ converges to $\mathbf{c}_{i,\infty}$ almost surely as $t \to \infty$, where

$$\begin{aligned} \mathbf{c}_{i,\infty} &= \frac{\mathbf{m}_{i,\infty}}{\gamma N_{i,\infty} + (1-\gamma) n_{i,\infty} + \epsilon} \cdot \frac{\sum_{i=1}^{K_m} (\gamma N_{i,\infty} + (1-\gamma) n_{i,\infty}) + K_m \cdot \epsilon}{\sum_{i=1}^{K_m} (\gamma N_{i,\infty} + (1-\gamma) n_{i,\infty})} \\ &= \frac{\ell_{i,\infty}}{n_{i,\infty} + \frac{\gamma\epsilon}{1-\gamma} + \epsilon} \frac{\sum_{i=1}^{K_m} (n_{i,\infty} + \frac{\gamma\epsilon}{1-\gamma} + \epsilon)}{\sum_{i=1}^{K_m} (n_{i,\infty} + \frac{\gamma\epsilon}{1-\gamma})}. \end{aligned} \tag{28}$$

The proof is complete. □

E Model Configurations

Our experiments use the base ViT architecture for audio and image, and the base video ViT architecture for video mode.

Audio and Image: For audio and image modalities, we follow the configurations of BEATs (Chen et al., 2023) and AudioMAE (Huang et al., 2022). The encoder architecture is based on a transformer model with appropriate modifications to handle mel-spectrograms for audio and patches for images.

Video: For the video modality, we adopt the setup from VideoMAE (Tong et al., 2022). The encoder is extended to process spatiotemporal data by incorporating temporal attention mechanisms to capture motion dynamics effectively.

The decoder is a 16-block Swin transformer for audio and image, and a 4-block transformer for video. The tokenizer estimator is a simple 3-block transformer for audio and image, and a 4-block transformer for video.

Each of $M = 4$ codebooks for RQ has a size of $K_m = 256$ code vectors of dimension $D = 256$, initialized based on the input latent vectors with k-means initialization of 10 steps, or with random uniform initialization. Meanwhile, the VQ version of the model has $M = 1$ codebook of $K_m = 1024$ code vectors also of dimension $D = 256$. Hence, the VQ and RQ versions of the training framework contains the same number of codes, where each latent vector is mapped to $M = 1$ code for VQ, and $M = 4$ codes for RQ.

F Benchmark Comparisons on ImageNet

Table 10: Fine–tuned ImageNet-1K accuracy for self-supervised models.

Method	Top-1 Acc (%)
ViT384-B-JFT300M (Dosovitskiy et al., 2021)	79.9
BRIDLE (ours)	81.1
DINO (Caron et al., 2021)	82.8
BEiT (Bao et al., 2022)	83.2
MoCo v3 (Chen et al., 2021)	83.2
MAE (He et al., 2022)	83.6

Although BRIDLE ranks below the masked-image-modeling family by 2–3 pp in Top-1 accuracy, its design offers several advantages that can make this trade-off attractive:

1. Training efficiency: BRIDLE reaches its reported score after only two bidirectional iterations, each markedly shorter than the 800–1600 epoch schedules typical for MAE or BEiT.

2. Discrete token output: Thanks to residual quantisation, BRIDLE supplies high-entropy visual tokens directly from the encoder, eliminating the need for a separate tokenizer when the features will feed downstream generative or retrieval models.

3. Modality-agnostic recipe: The same encoder–tokenizer framework has been shown to transfer to audio and video without architectural changes, yielding consistent gains over VQ baselines across modalities.

4. Label-efficient features: In linear probing, BRIDLE outperforms its VQ counterpart, indicating that its representations capture semantically useful structure even when only a small fraction of ImageNet labels is available.

Overall, BRIDLE offers compute-conscious pre-training that natively outputs discrete tokens, making it an appealing foundation when cross-modal alignment, low-resource fine-tuning, or downstream generative tasks.

G Codebook and Tokenizer Evaluation

To evaluate the effectiveness of the codebooks in the BRIDLE model's tokenizer, we use several metrics that assess both the diversity of code usage and the balance of their usage. These metrics help ensure that the tokenizer effectively utilizes the entire codebook, leading to robust audio representations.

Code Usage Rate (CUR): Measures the proportion of codes within every codebook $\mathcal{C}_m$ that are used at least once during tokenization:

$$\text{CUR} = \frac{\text{Number of unique codes used}}{K_m}. \tag{29}$$

A high CUR value indicates diverse usage of the codebook.

Usage Entropy (UE): Quantifies the uniformity of code usage using Shannon entropy (Shannon, 1948):

$$\text{UE} = -\sum_{i=1}^{K_m} p_i \log(p_i), \tag{30}$$

where $p_i = \frac{\hat{N}_{i,t}}{\sum_{j=1}^{K_m} \hat{N}_{j,t}}$ is the probability of selecting the i-th code, with $\hat{N}_{i,t}$ being the cluster size of the i-th code across the entire dataset. A higher UE suggests more balanced code usage.

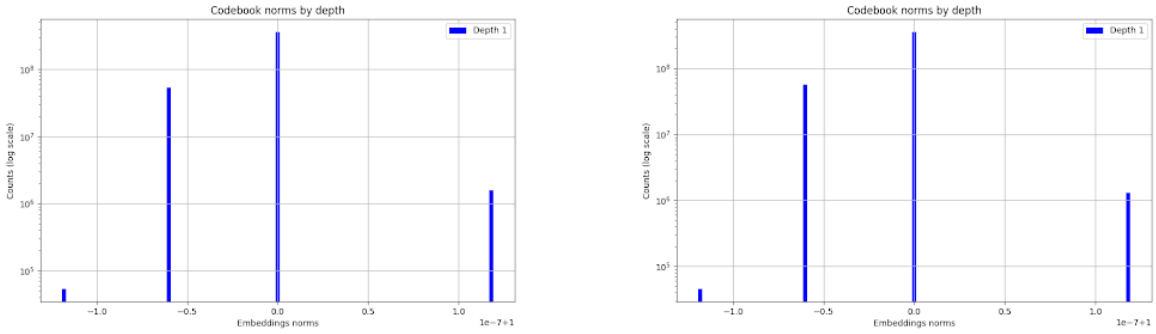
Effective Code Usage (ECU): Combines CUR (29) and UE (30) to evaluate both the diversity and balance of code usage:

$$\text{ECU} = \text{CUR} \times \frac{\text{UE}}{\log(K_m)}. \tag{31}$$

In Table 11, we show ∼100% CUR for pre and post-training RQ. Additionally, the table shows superior ECU of RQ compared to VQ. A well-spread distributions of code usage is also demonstrated in Figure 3 for RQ codebooks.

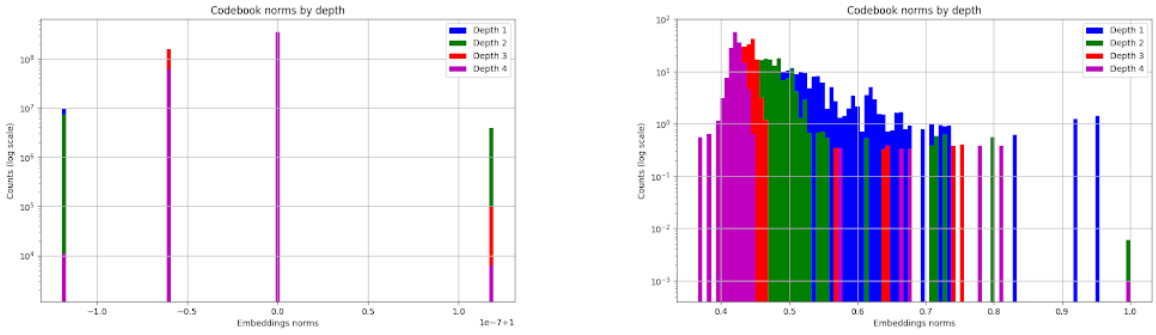
Table 11: CUR and ECU for different methods before and after training

Iteration	Method	Codebook #	CUR ↑	ECU ↑
$\text{BRIDLE}_{\text{iter1}}$ (random init codebook)	VQ	-	1.00	0.0146
	RQ (uniform)	1	1.00	0.0235
		2	0.99	0.0284
		3	1.00	0.0489
		4	0.86	0.0367
	RQ (k-means)	1	0.99	0.0234
		2	1.00	0.0326
		3	1.00	0.0320
		4	1.00	0.0310
$\text{BRIDLE}_{\text{iter2}}$ (post training codebook)	VQ	-	0.21	0.0041
	RQ (uniform)	1	0.99	0.0275
		2	1.00	0.0231
		3	1.00	0.0228
		4	1.00	0.0229
	RQ (k-means)	1	0.99	0.0283
		2	1.00	0.0229
		3	1.00	0.0228
		4	1.00	0.0228



(a) (VQ) $\text{BRIDLE}_{\text{iter1}}$ (b) (VQ) $\text{BRIDLE}_{\text{iter2}}$

Figure 2: VQ codebook embeddings norm distributions before and after training on AudioSet



(a) $\text{BRIDLE}_{\text{iter1}}$ (b) $\text{BRIDLE}_{\text{iter2}}$

Figure 3: RQ codebook embeddings norm distributions before and after training on AudioSet