Codified Audio Language Modeling applied to Instrument Recognition

Marco Conati
Harvey Mudd College
mconati@g.hmc.edu

Alec Vercruysse
Harvey Mudd College
avercruysse@g.hmc.edu

***Abstract*—Building upon the work of Castellon, Donahue, and Liang [1], we demonstrate that codified audio language modeling can learn useful representations for instrument identification. To this end, we used representations from layers within Jukebox: a generative model for music [2]. Jukebox representations are used as input features to train shallow probes; each probe is trained to identify the presence of a specific instrument. As input data, the OpenMIC-2018 dataset was used [3]. This dataset contains 10 second excerpts of audio that are partially labeled for the presence or absence of 20 instrument classes. Our results indicated that our probes were unable to match state-of-the-art performance. Further investigation also revealed that while some of the best performing models were achieved with few to no hidden layers, the probes still learn a complicated representation of Jukebox output features.**

I. Introduction

Humans are fairly adept at discerning instruments in complex pieces of music. Musical instrument recognition is the task of automatizing this skill and is an active area of research within the field of Music Information Retrieval (MIR). While instrument recognition has achieved success in recognizing isolated instruments, recognizing instruments in music - where multiple instruments are played at once - remains a difficult task. This task is notably difficult due to superposition of sources, large timbre variations within individual instruments, and the lack of data for supervised learning [4].

As mentioned above, the lack of data for supervised learning is a primary challenge in the instrument recognition task. Datasets for polyphonic music are either strongly or weakly labeled. Strongly labeled datasets contain information about the onset and offset times of each instrument, while weakly labeled datasets only have information about the presence of an instrument in a clip, even if it is inconsistently active during the recording. Due to the added cost of annotation, strongly labeled datasets scale poorly in comparison to weakly labeled datasets.

OpenMIC 2018, the dataset used in this paper, contains 20,000 10s clips from songs across various genres [3]. Despite being weakly labeled, this dataset was selected as it contains a large sample size as well as a uniform distribution across instruments.

Music instrument recognition has a variety of use cases. A reliable recognition system would be helpful for instrument retrieval systems, allowing individuals to search for music based on instrumentation. Within MIR, instrument recognition could be useful for a variety of tasks. For example, instrument tasks could assist in identifying affinities towards certain instruments, automatic transcription, and source separation [4].

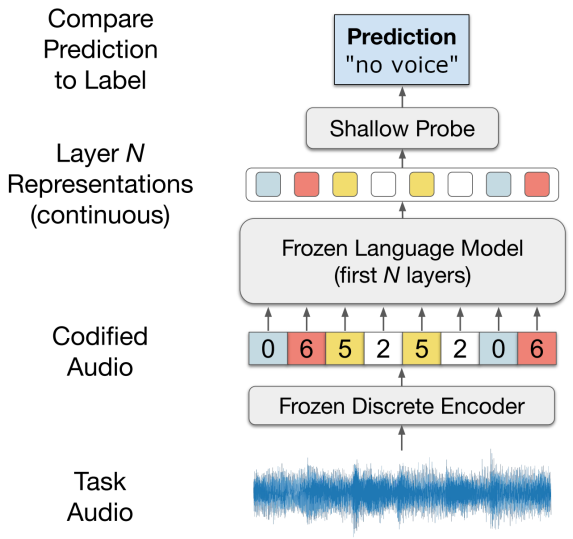

Fig. 1. Block diagram of system design, adapted from Castellon et. al. [1]. The VQ-VAE used as the frozen discrete encoder and the first N layers of the sparse transformer used as the frozen language model were pretrained Jukebox models. The shallow probe was developed for instrument recognition on the OpenMIC-2018 dataset using the output of the language model.

Previous approaches to music instrument recognition have focused on building deep neural networks from the ground up [4], [5]. Recent state-of-the art approaches have used CNN models to extract audio features and attention based LSTMs to learn temporal features [4], [5]. Our approach is different in that it utilizes transfer learning. CALM representations are extracted from the Jukebox model, and training is only performed on a shallow probe.

II. System Design

Instrument recognition probes were trained via transfer learning, using a codified audio language model (CALM) to compute the feature representations of the input data. Our approach to transfer learning with CALM representations was adapted from Castellon et. al. [1]. The Jukemir github repos-

itory was used as starter code for interfacing with Jukebox and guide for probe design. For our project, the Jukemir code was modified to interface with the OpenMIC dataset and new shallow probes were iteratively trained following the Jukemir grid search procedure. The block diagram of the system can be seen in Figure 1.

A. Data Considerations

OpenMIC was selected for use in training as it contains a large sample size from across various genres with a uniform distribution of instruments. However, the dataset also presents challenges in its format as not all clips are labeled for all 20 instruments. This complicates training if models are to predict the presence or absence of all instruments on a single clip. To address this issue, the dataset was split into 20 partitions corresponding to the various instruments where all samples in each partition had a label for that instrument. Individual models were then trained on each partition, so that each model was predicting the presence or absence of a single instrument. To predict the presence/absence of all instruments in a single clip, a function was created to pass CALM Representations through each model.

An additional challenge resulted from the difference in input samples between the original Jukebox model and OpenMIC. Jukebox expects 62 second audio clips for feature extraction, while the samples in OpenMIC were only 10 seconds long. Castellon et al. [1] had done considerable work to tune Jukebox hyperparameters and identify which layer features were useful for MIR tasks. Keeping this in mind, we opted to modify our audio so that we could maintain all other parameters that had demonstrated good performance. Two approaches were investigated to make OpenMIC audio clips match the Jukebox input size: upsampling the audio and concatenating samples. In the upsampling approach, the audio files were loaded with a sample rate 6.2 times greater than the value used by Castellon et al., for a sample rate of approximately 273 kHz [1]. In the concatenation approach, OpenMIC samples were concatenated to themselves 6.2 times to create 62 second audio clips.

B. CALM Representations

As input features for training probes, Codified Audio Language Model (CALM) feature representations were computed from each OpenMIC sample using the method outlined in [1].

Recent works have attempted to represent raw-audio in the time domain as a set of symbols akin to a language. This enables the use of state-of-the art natural language models, which have shown to succeed at discriminative tasks, to be used in music information retrieval tasks [1]. CALM features are the outputs of these natural language models.

Since traditional audio is generally sampled at the order of 44 kHz, and each sample is 16 bits, the space of traditional audio clips is extremely large. Encoding chunks of raw audio into discrete ”codewords” helps reduce this dimensionality while mitigating information lost.

In the pre-trained model, a Vector Quantized Variational Autoencoder (VQ-VAE) has trained to produce this low-dimensional feature representation. VQ-VAEs are similar to variational autoencoders, which attempt to learn a low-dimensional latent space with enforced priors to generate ensure semantically meaningful encodings. VQ-VAEs use a discrete ”codebook”, however, of vectors, instead of using a continuous latent space. For example, the VQ-VAEs trained to provide low-dimensional representations as inputs to the language model used a codebook size of 2048. This codebook was trained using gradient descent alongside the rest of the autoencoder, and when the autoencoder is at the discretization step, the discretized vector is selected by choosing the codeword with the lowest mean squared error [2].

The language model used was the hierarchical sparse transformer model used in Jukebox [2]. Several VQ-VAEs trained at various hop-sizes were used as inputs to the language model in order to keep track of long-term dependencies in the audio. The middle layer of the transformer, layer 36, had previously been found to be the one that worked best for general MIR tasks, so the activations at this layer were chosen to be the CALM features [1]. In addition, previous work also reported success lowering the dimensionality of the CALM features by mean pooling across time, which was crucial to reducing the extremely high dimensional output of the natural language model while not loosing too much information [1]. The output of the language model was a continuous vector of size 4800.

C. Probe Design

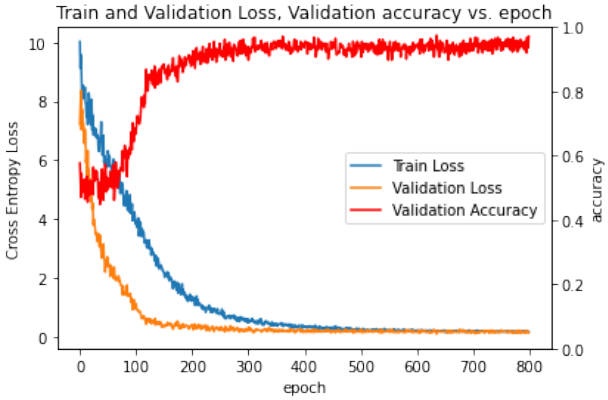

Fig. 2. Training Curve for One-Layer Voice Recognition Probe

Shallow, fully connected models were trained to probe CALM representations for relevant information regarding instrument recognition. The design of the probes was iteratively determined for each instrument using a grid search over the following hyperparameters:

- Model: One Linear Layer,
  One Layer MLP with 512 hidden units,
  Two Layer MLP with [512, 256] hidden units
- Dropout: 0.25, 0.5, 0.75
- Regularization: 0, 1e-4, 1e-3

This grid search was performed for each instrument probe (20 instruments) on all configurations (27 configurations) for a total of 540 iterations. The optimizer was set as ADAM with learning rate 1e-4. Each configuration was trained for 800 epochs. Performance of each probe was optimized by minimizing binary-cross-entropy loss, and the probe design with minimal validation loss for each instrument was selected.

III. Results

An example training curve is shown in Figure 2 for the one-layer voice recognition probe. All probe configurations converged during the 800 epochs of training.

As seen in Table I, the performance of the model varied widely across different instruments. Interestingly, the model performed best on the most popular instruments, such as piano, guitar, voice, and drums, even though the dataset was evenly tagged for every instrument. In every model with an F1 score above .90, each model used only a single hidden layer, and 1e-4 regularization. The ideal dropout varied.

The worst result by far was for the clarinet model, however a number of other models also performed poorly.

The best model, piano, displayed a test F1 score that is similar to recent purpose-built models for instrument recognition on the OpenMIC dataset [4], [5]. On average, however, CALM-based models performed less accurately than the state-of-the-art purpose built models. The relative performance of each instrument is similar to the relative performance of the different instruments in purpose-built models. Notable exceptions from the high performance instruments are the drums and the cymbals. Particularly, Anhar and Gururani et. al. both report models that perform significantly better on cymbal classification than drum classification. The drum classification of our CALM-based model, however, outperformed not only the cymbal classification of our model, but also the other two models' drum classifications [4], [5].

Using concatenation as a method of formatting the 10 second OpenMIC audio clips to be the right size for the encoding and NLP model showed consistently improved results over upsampling the audio to increase the number of samples. The test F1 score of every single model was improved by using concatenation over upsampling. On average, an F1 score rose about 3.9 percentage points, for a mean 7% improvement in F1.

IV. Discussion

A. Model Predictions

In calculating F1 scores, a pattern emerged between unsuccessful probes. Probes with the lowest F1 score, like clarinet, ukulele, and mandolin, tended to suffer from low precision. Further examination of the training data for these instruments revealed that these instruments were comparatively uncommon in labeled samples. On one hand, clarinet, ukulele, and mandolin were only present in 6.8, 12.9, and 14.9 percent of labeled samples respectively. On the other hand, piano, guitar, and voice were present in 60.9, 61.6, and 55.9 percent of samples respectively. This lack of representation could have contributed to difficulties in learning instrument characteristics. This conclusion is also supported by the probe predictions. For well represented instruments, the probes made high confidence predictions about the presence of instruments (Figure 3) while for poorly represented instruments, the confidence of predictions was considerably lower (Figure 4).

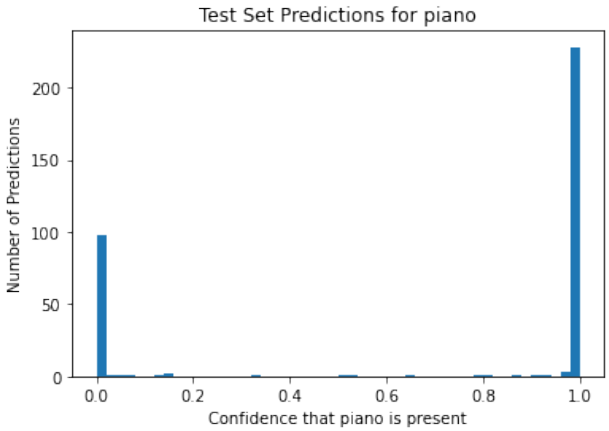

Fig. 3. Histogram of Predicted Probabilities on the Test Set for piano

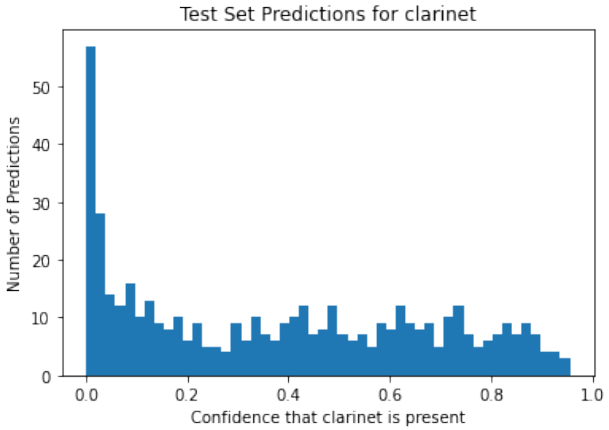

Fig. 4. Histogram of Predicted Probabilities on the Test Set for clarinet

Note that during training, the sampling was weighted to draw an equal amount of positive and negative samples for each instrument. This was done in order to improve recall, and accounts for the false-positive predictions on the test data. Ultimately, more samples, or more even weighting of samples, are needed to expect to achieve similar results on the low-performing instruments as the high-performing instruments.

B. Model Weights

The majority of our probes were most successful with a zero layer model. These models represent a situation where CALM representations are directly weighted to achieve our output prediction. With this insight, the parameters of zero layer probes were plotted in an effort to identify what CALM representations yielded useful information in image recognition. An example plot is shown in Figure 5.

TABLE I
GRID SEARCH RESULTS: CONCATENATED INPUT AUDIO

Instrument	Val. Loss	Val. Acc.	H. Layers	Reg.	Dropout	Test F1
piano	0.111	0.968	[512]	1e-4	0.25	0.924
guitar	0.168	0.964	[512]	1e-4	0.75	0.919
voice	0.104	0.974	[512]	1e-4	0.50	0.915
drums	0.125	0.963	[512]	1e-3	0.25	0.912
synthesizer	0.198	0.956	[]	1e-4	0.25	0.886
cymbals	0.180	0.954	[]	1e-3	0.50	0.864
violin	0.264	0.897	[]	1e-4	0.50	0.762
saxophone	0.289	0.907	[]	0	0.25	0.681
trumpet	0.334	0.902	[]	1e-4	0.75	0.641
cello	0.328	0.902	[]	1e-3	0.75	0.608
organ	0.354	0.865	[]	0	0.75	0.604
mallet_percussion	0.322	0.867	[]	0	0.75	0.584
trombone	0.379	0.848	[]	1e-3	0.75	0.531
flute	0.466	0.772	[]	1e-3	0.25	0.505
banjo	0.377	0.856	[]	1e-3	0.25	0.494
bass	0.366	0.857	[]	1e-3	0.50	0.477
accordion	0.381	0.850	[512, 256]	1e-3	0.25	0.476
mandolin	0.451	0.809	[]	1e-3	0.25	0.445
ukulele	0.405	0.816	[512, 256]	1e-4	0.25	0.437
clarinet	0.464	0.776	[]	1e-4	0.25	0.162

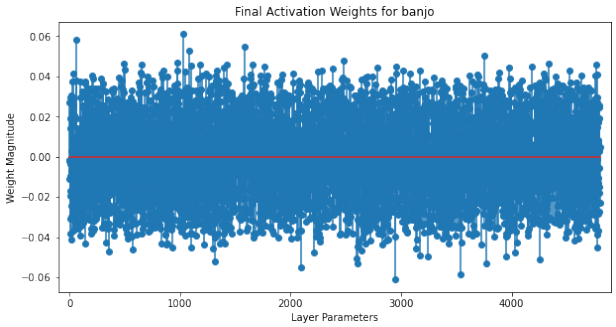

Fig. 5. Output weights of the zero-layer probe for recognizing banjo

Weights for zero-layer probes trained on other instruments had a similar shape. There does not appear to be any specific region of the CALM representation that is particularly relevant to instrument recognition, and while the model is making a prediction based off a linear combination of NLP output features, it still seems to learn a complicated representation of these features.

C. Upsampling vs. Concatenation

The improvement in performance seen when using concatenation as a method instead of upsampling makes sense considering the encoding scheme of the VQ-VAE. Passing in audio with a fixed hop size (in samples) but increasing its sample rate makes the autoencoder compute each latent vector over a smaller region of time, effectively decreasing hop-size. Concatenating audio together, however, to form a longer sample, does not have any effects on the hyperparameters used when encoding the song and performing NLP. For this reason, it makes sense that concatenation of audio results in latent space encodings that are more informative to the probes.

V. CONCLUSION

Ultimately, while our our CALM-based approach was relatively successful at instrument recognition, it was unable to match the performance of state-of-the art methods. Notably, while performance lagged behind state-of-the-art approaches, training was still considerably easier as this approach utilized transfer learning and lightweight probes.

Future work could include investigating the temporal characteristics of CALM features. CALM-based models have demonstrated success in genre classification, emotion recognition, and key detection. However, these tasks differ from instrument recognition in that they are relatively unchanging over audio recordings; while instruments vary in activity during different parts of recording, genre, key, and emotion are more stable. As our approach built off successful CALM-based models, we adopted their method of mean-pooling CALM representations over time to reduce their dimensionality (this is necessary as the full representation yields over 10 GB of data for a 24 second audio clip). This approach may not be advantageous for instrument recognition, as information about instruments present in a small portion of a recording may be lost during averaging. This idea is supported by the approaches of state-of-the-art methods. State-of-the-art methods use attention mechanisms so that the model can focus on regions of the input where target instruments are present. Future work could investigate mean-pooling smaller sections of the CALM representations to maintain some temporal characteristics and allow for attention-based probes.

VI. ACKNOWLEDGEMENT

We would like to thank Prof. TJ Tsai for his guidance with this project.

REFERENCES

[1] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," *CoRR*, vol. abs/2107.05677, 2021. [Online]. Available: https://arxiv.org/abs/2107.05677

[2] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," 2020.

[3] E. J. Humphrey, S. Durand, and B. McFee, "Openmic-2018," Sep. 2018. [Online]. Available: https://doi.org/10.5281/zenodo.1432913

[4] S. Gururani, M. Sharma, and A. Lerch, "An attention mechanism for musical instrument recognition," 2019.

[5] A. K. Anhari, "Learning multi-instrument classification with partial labels," 2020.