Given a set of music recordings and the associated text elements for each recording, we construct a cross-modal training dataset of (audio, text) pairs as follows. For each recording, we compute an F-channel log mel spectrogram and extract a collection of T-frame context windows. We null-pad or truncate each associated text element to a fixed length n. Then, each mini-batch $\mathcal{B}$ consists of a set of B target audio-text pairs of the form $\{(\mathbf{x}^{(i)}, \mathbf{t}^{(i)})\}_{i=1}^{B}$. Here, each target pair is sampled by first selecting a random recording and sample a random spectrogram context window $\mathbf{x}^{(i)} \in \mathbb{R}^{F\times T}$ from it. Next, we randomly select one of its associated text elements $\mathbf{t}^{(i)} \in \mathcal{A}^n$. This sampling scheme means that multiple epochs are required to cover the entirety of the training audio and all the associated text. We also experimented with concatenating multiple text annotations for each example, but it did not generally work as well.

We train to minimize a batch-wise Contrastive Multiview Coding loss function [33], which is a cross-modal extension of the popular InfoNCE and NT-Xent losses [34, 35]. For each batch $\mathcal{B}$, this loss $\mathcal{L}(\mathcal{B})$ takes the form

$$\sum_{i=1}^{B} -\log\left[\frac{h[f(\mathbf{x}^{(i)}), g(\mathbf{t}^{(i)})]}{\sum_{j\neq i} h[f(\mathbf{x}^{(i)}), g(\mathbf{t}^{(j)})] + h[f(\mathbf{x}^{(j)}), g(\mathbf{t}^{(i)})]}\right],$$

where h is a critic function given by $h[\mathbf{a}, \mathbf{b}] = \exp(\mathbf{a}^T\mathbf{b}/\tau)$ for $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$, and $\tau \in (0, 1]$ is a trainable temperature hyperparameter. For our ℓ_2-normalized embedding model outputs, the inner product is effectively cosine similarity. The critic's goal is to produce a large positive value for target audio-text pairs, and a small value close to zero for all non-target pairs constructed within the batch. Temperature values less than one function to increase the output range of h. Previous research [35, 36] demonstrated that a large batch size is beneficial to contrastive loss optimization.

3.2 Audio Embedding Network

For the audio embedding tower, f, we consider two proven audio architectures. Following its introduction to the audio machine learning community [37], the *Resnet-50* architecture has become a common and well-performing option. It is a straightforward adaptation of the original vision architecture: as in [37], we remove the stride of 2 in the first convolutional layer and apply to log mel spectrograms ($F = 64$ mel channels, 25 ms Hanning window, 10 ms step size) treated as grayscale images. Unlike the Resnet-50 model in [37] which operated on 0.96-second context windows, in order to allow the modeling of longer-term musical structure, our implementation takes as input 10-second windows (randomly selected from each training clip), in the form of $(F = 64) \times (T = 1000)$ spectrogram patches. During training, we apply SpecAugment to each spectrogram using the parameters from [10] before passing it into the embedding network. A final mean pooling operation is applied across time and mel channels, followed by a linear fully connected layer with $d = 128$ units, whose output is ℓ_2-normalized. We pretrain all but the final linear transform layer via logistic regression on AudioSet [17], including all 527 classes, and discard the final classifier layer before fine-tuning for our task.

Table 1. Text annotation examples.

Type	Examples
Short-form (SF)	tags like genre, mood, instrument, artist name, song title, album name
Long-form (LF)	'Hip-hop features rap with an electronic backing.' 'The melody is so nostalgic and unforgettable.'
Playlist (PL)	'Feel-good mandopop indie', 'Latin workout' 'Salsa for broken hearts', 'Piano for study'

Audio Spectrogram Transformer (AST) is a port of the successful Vision Transformer (ViT) base architecture and is currently the state-of-the-art in the audio event classification space [10]. AST consists of a stack of 12 Transformer blocks (hidden dimension 768, 12 self-attention heads) that are applied to a sequence of "tokens" corresponding to a flattened set of linear-transformed 16×16 (stride 10 along both axes) time-frequency patches extracted from the $(F = 128) \times (T = 1000)$ log mel spectrogram context windows. We again apply SpecAugment during training. Similar to the Transformer-based language models, trainable positional encodings are added to the sequence of patch tokens, and a `[CLS]` token is prepended to the sequence as a summary of the contextual patch embeddings. We apply a linear fully-connected layer with $d = 128$ units and ℓ_2-normalization to the final 768-dimensional encoding at the `[CLS]` token position, and this forms the output of audio embedding network f. We warm-start training for all but the final linear transform layer using the public AST checkpoint [10].

3.3 Text Embedding Network

For the text embedding model, we consider the commonly-used Bidirectional Encoder Transformer (BERT) with base-uncased architecture [38], which consists of a stack of 12 Transformer blocks (hidden dimension of 768 and 12 self-attention heads). We apply the BERT wordpiece tokenizer to convert a text input string into a sequence of tokens ($n = 512$). The output of the text embedding network is defined to be the `[CLS]` token embedding, linearly transformed to the shared audio-text embedding space of dimension $d = 128$ and subsequently ℓ_2-normalized. We warm-start our text embedding network using the publicly available checkpoint [39].

3.4 Training Dataset Mining

To assemble a large-scale collection of (audio, text) pairs needed to train our MuLan embedding models, we start with a collection of 50 million internet music videos. From the soundtrack of each video, we extract a 30-second clip starting at the 30 second mark. We then apply a pre-existing music audio detector and discard any clip that is less than half music content. After this filtering, we are left with approximately 44 million 30-second clips, which amounts to nearly 370K hours of audio.

Table 2. Statistics for text data sources. Tokens counts (in billions) are across all 44M videos. APV is the average number of text annotations (i.e. separate free-form strings) per video, including those with none.

	Pre-filter		**Post-filter**	
Type	Tokens (B)	APV	Tokens (B)	APV
Short-form	31.2	42.9	5.4	29.6
Long-form	30.7	70.7	0.2	0.4
Playlists	2.5	24.3	-	-

For each music video, we consider 3 sources of noisy text data: (i) *short-form* (SF) text including video titles and tags; (ii) *long-form* (LF) text including video descriptions and comments; and (iii) titles of 171 million *playlists* (PL) that are linked to the internet music videos in our dataset. None of these text sources is guaranteed to be referring to the musical properties of the soundtrack. In particular, comments data contains the most noise, and can be subjective or less directly related to the music content. In Table 1, we show examples that are indeed music-related to give the readers a flavor of each type of text annotation.

In observance of the highly noisy text, we experimented with training MuLan with the SF and LF text data filtered to a cleaner set of music-descriptive annotations (PL is used unfiltered). For this, we fine-tune a pre-trained BERT model with a binary classification task on a small curated set of 700 sentences, which are manually labeled to be music-descriptive or not. We then apply this text classifier to filter the sentences in the LF annotations. Separately, we apply a set of rule-based filtering heuristics to clean up the SF annotations. Table 2 shows the size and coverage of each of these text sources, both before and after filtering. Note that playlist titles and filtered long-form annotations are only available for a minority of recordings in the dataset (18M and 6.8M out of the total 44M, respectively).

We also convert AudioSet into a set of audio-text pairs, denoted below as ASET. Specifically, we include all examples for all 527 classes, using each label string attached to an example as an associated text annotation. This results in a set of approximately 2 million 10-second clips for training, each with 1.8 label annotations on average. Given the great scale imbalance of these four different data sources, which is often at odds with their linguistic richness and quality, we construct each mini-batch with a prescribed set of proportions that were chosen without any optimization: 2:2:1:1 for SF:LF:PL:ASET. This means that despite its small scale, the (e.g.) filtered LF annotations still comprise 1/3 of each mini-batch.

4. EXPERIMENTS

We evaluate MuLan using both the Resnet-50 audio encoder (M-Resnet-50) and AST audio encoder (M-AST). In both cases we use the BERT-base-uncased architecture as the text encoder. We train all models for 14 epochs on the collection of audio-text pairs mined from the 44M music recordings and the processed text labels in all categories: AudioSet (ASET), short-form tags (SF), long-form sentences (LF), playlist information (PL). We use the Adam optimizer with a step decay learning rate schedule using a decay factor 0.9 applied every 40K steps and initial values of 5×10^{-5} for M-Resnet-50 and 4×10^{-5} for M-AST. The temperature parameter is initialized to $\tau = 0.1$ for all models. M-Resnet-50 is trained with a batch size of $B = 6144$ pairs, while $B = 5120$ pairs were used for M-AST due to memory limitations. Since M-AST and M-Resnet-50 show roughly similar performance in the evaluation tasks considered, we use M-Resnet-50 throughout the text ablation study for its better training efficiency.

4.1 Evaluation Tasks

4.1.1 Zero-shot Music Tagging

Given a music clip and a set of candidate text label tags, we define each prediction score as the cosine similarity between the audio embedding of the music clip and the text embedding of each tag string. The generalization ability of the proposed method to potentially unseen target labels is achieved through (i) the use of a contextual text encoder, which provides a flexible prediction space, and (ii) the use of cross-modal contrastive learning to anchor the language semantics to an audio representation.

We conduct this evaluation with two music tagging benchmarks: MagnaTagATune (MTAT) [40] and the music related portion of AudioSet [17]. For MagnaTagATune, we consider both the well-exercised top-50 tag set, as well as the full 188 tag set. We use standard train/validation/test partitions (note that zero-shot experiments do not use train/validation) and report class-balanced area under the receiver operating characteristic curve (AUC-ROC) on the test set. The audio clips in MagnaTagATune are 29 seconds long, so we split each into 3 non-overlapping 10-second segments and average the segment-level embeddings to get the clip-level embedding. For AudioSet, we consider a 25-way genre tagging task (Gen-25) as studied in [25], and a richer 141-way tagging task (Mu-141) that includes the entire music subtree of AudioSet ontology.

It is important to note that AudioSet is included in contrastive training, and a fraction of MTAT classes overlap with the AudioSet ontology. As a result, AudioSet and (to lesser extent) MTAT evaluations are not strictly zero-shot from a label exposure perspective. However, the explicit, matched AudioSet supervision is diluted by the abundance of free-form language supervision during MuLan training. Therefore, by comparing MuLan models and conventional AudioSet classifiers, we can measure the cost of moving to a flexible natural language interface that additionally supports classes outside the AudioSet ontology.

4.1.2 Transfer Learning with Linear Probes

In addition to the zero-shot experiments introduced above, we also evaluate the audio encoder as a general purpose feature extractor for downstream tagging tasks. We again consider the two benchmarks of MagnaTagATune and AudioSet, and use the training datasets to train an independent per-class logistic regression layer on top of the frozen 128-dimensional audio embeddings. We follow the same eval-

Table 3. Text triplet evaluation examples.

Eval Set	Anchor / Positive / Negative
Ontology	Steelpan / Sounds of a tuned percussion instrument originally constructed from steel oil drums by hammering out small patches on the head to produce separate pitches. / The sound of a musical instrument that produces sound by vibration of air in a tubular resonator in sympathy with the vibration of the player's lips.
Playlist	Relaxing Korean Pop / Lets make your chill mood with a collection of easy-going sounds from Korean artists. / These fun and upbeat songs from the alternative side of the pop music spectrum will keep you energized while you exercise.

uation protocol of past transfer learning studies using these datasets, allowing for a direct comparison of performance.

4.1.3 Music Retrieval from Text Queries

Given a music search collection and a text query, MuLan provides the ability to retrieve the music clips that are closest to the query in the embedding space. This evaluation is relevant to music retrieval applications, where content features can offer finer-grained and more complete similarity information when compared with metadata-based methods [41]. We consider a proprietary collection of 7,000 expert-curated playlists, which do not overlap with the playlist information used in training. Each expert-curated playlist has a title and a description, and consists of 10-100 music recordings. The playlist titles are usually short phrases, including a mixture of genres, sub-genres, moods, activities, artist names, and compositional elements (e.g. *'Indie Pop Workout'*, *'Relaxing Korean Pop'*). Playlist descriptions consist of one or more complete sentences (see pos/neg entries of "Playlist" row of Table 3 for examples). The playlist evaluation includes approximately 100K unique recordings.

We construct two cross-modal retrieval evaluation sets from the expert-curated playlist data, one using titles as queries and the other using descriptions. For each dataset, we use the recordings belonging to the corresponding playlist as the ground truth retrieval targets, and all the 100K recordings as the pool of candidates. We report both AUC-ROC and mean average precision (mAP). We use the same embedding averaging and cosine similarity-based scoring mechanism as in the zero-shot tagging case. However, the playlist information is of substantially different nature compared to the tags involved in the music tagging benchmarks. Instead of a small vocabulary of mostly basic genres and instruments, the playlist titles and descriptions have much finer-grained information and are similar to queries that are presented to music search engines.

4.1.4 Text Triplet Classification

Compared to the conventional pre-trained BERT model, our text encoder is fine-tuned using in-domain music data and cross-modal contrastive loss. Note that there are no text-only training objectives. To measure whether our proposed method deepens the text encoder's understanding of

Table 4. Music tagging results reported in AUC-ROC.

Model	AudioSet Gen-25	AudioSet Mu-141	MTAT Top-50	MTAT All-188
(a) Zero-shot (Trained w/ ASET + SF + LF + PL)				
M-AST	0.840	0.909	0.778	**0.776**
M-Resnet-50	0.840	0.899	**0.782**	0.772
(b) Text ablation (using M-Resnet-50 Zero-shot)				
ASET + SF + LF	0.839	0.907	0.760	0.756
ASET + SF	0.839	0.885	0.754	0.747
ASET	**0.886**	**0.942**	0.753	0.771
SF/LF Unfiltered	0.845	0.908	0.774	0.766
(c) Linear probe				
M-AST	0.906	**0.942**	0.925	0.953
M-Resnet-50	**0.910**	0.940	**0.927**	**0.954**
Baselines:				
Hybrid [25]	0.904	0.920	0.915	0.941
JukeBox [15, 23]	-	-	0.915*	-
MuLaP [32]	-	-	0.893*	-
CLMR [22]	-	-	0.866*	-
(d) End-to-end training baselines				
AST [10]	0.888	**0.949**	-	-
SC-CNN [42]	-	-	0.913*	-

* indicates that the number is brought from the original paper.

music related text, we directly evaluate the text embeddings with a triplet classification task. Each triplet consists of 3 text strings of the form of (*anchor*, *pos*, *neg*), and it is considered correct if *pos* is closer than *neg* to *anchor* in the text embedding space. We derive two such text triplet evaluation sets. The first uses the AudioSet ontology [17]: for each of the 141 music related classes, we use its label string as the anchor text, its long-form description as the positive text, and sample 5 random class's long-form description as the negative text to construct 5 triplets. For the second set, we sample 1,000 triplets from the expert-curated playlist data in a similar fashion: we first sample a playlist, set the anchor and positive text to be its title and description, respectively, and then set the negative text to be the description of another randomly sampled playlist. Examples of both sets are shown in Table 3.

4.2 Results and Discussion

4.2.1 Music Tagging

Table 4(a) shows the zero-shot tagging metrics, where M-Resnet-50 and M-AST obtain comparable performance. Note that there can be a significant misalignment between the word sense of a label in the tagging evaluation compared to that in our training text. This can lead to a degradation in performance relative to the explicitly supervised linear probe setting where the task-expected tag semantics can be learned. The MTAT gap is substantially larger than AudioSet's, driven by particularly bad performance for (i) MTAT tags with nonspecific meaning or multiple senses, e.g. "weird" and "beats"; and (ii) MTAT tags involving simple negation (e.g. "not rock", "no piano"). This is a result of the text encoder not adequarely modeling the meaning of these negated concepts, which is a well known problem with BERT [43, 44] (the text embedding of "not rock" is similar to "rock" and performance suffers).

Table 4(b) shows the results of the text ablation study,

Table 5. Text query music retrieval evaluation results. Text ablation/unfiltered models use M-Resnet-50.

	Title		Description	
Model	AUC	mAP	AUC	mAP
M-AST	**0.933**	0.110	**0.903**	**0.090**
M-Resnet-50	0.931	0.104	0.901	0.084
Text Ablation:				
ASET+SF+LF	0.917	0.101	0.892	0.077
ASET+SF	0.913	0.089	0.867	0.060
ASET	0.626	0.005	0.688	0.009
SF/LF Unfiltered	**0.933**	**0.111**	0.897	0.081

Table 6. Text triplet classification accuracy AudioSet ontology evaluation and Playlist title to description evaluation. Text ablation/unfiltered models use M-Resnet-50.

Model	Playlist	AudioSet
M-AST	**0.959**	**0.962**
M-Resnet-50	0.945	0.951
Text Ablation:		
ASET + SF + LF	0.935	0.952
ASET + SF	0.910	0.938
ASET	0.693	0.818
SF/LF Unfiltered	0.949	0.959
Baselines:		
SimCSE [45]	**0.950**	0.938
SBERT [46]	0.942	0.889
USE [47]	0.918	**0.946**
BERT [38]	0.850	0.847

which aims to understand the benefits of different sources of text labels. Note that as we remove each dataset we maintain the same proportions described in Section 3.4. Unsurprisingly, training with AudioSet alone gets the highest AUC in AudioSet evaluation, with the text encoder learning the exact label semantics reflected in the test data. On the other hand, including more data sources in general improves performance on all other downstream tasks (MTAT, retrieval/text triplet evaluations in Tables 5 and 6) and the loss on AudioSet AUC is relatively minor. We observe that for the music tagging tasks considered, training with unfiltered data actually achieves comparable performance compared to the filtered version. That the model learns similarly useful associations without being overwhelmed by the sheer amount of noise in the raw text data came as a surprise. We speculate that our text filtering was too aggressive, having removed annotations that were not obviously music-related, but semantically important nonetheless. Since contrastive learning is highly noise tolerant, the gain from restricting to more strongly aligned audio-text pairs may have been offset by the loss of a large set of additional useful pairs.

Table 4(c) shows that when applying linear probes on MuLan audio embeddings, we achieve SOTA transfer learning performance on all tagging tasks. This demonstrates that MuLan's pretrained audio encoder continues to produce high quality general-purpose music audio embeddings, while also supporting new natural language applications. Finally, Table 4(d) lists end-to-end training baselines for 3 of these tasks. Our linear probe results exceed 2 of 3, and only slightly trails a SOTA AST AudioSet classifier.

4.2.2 Music Retrieval from Text Queries

In Table 5, we evaluate MuLan models (including with text/filter ablation) on the query retrieval evaluation tasks introduced in Section 4.1.3. Even though we start with a BERT checkpoint pretrained with massive language resources, training MuLan with only AudioSet clips and label annotations provides very limited ability to ground in-domain natural language to music. Such limited cross-modal supervision does not generalize to the rich semantics that appear in the playlist titles and descriptions, which are more in line with the complex queries that are presented to real-world music search engines. We observe significant gain after including the large-scale short-form tags mined from the internet, which helps the model learn to ground more fine-grained music concepts. There is additional gain when including comments and playlist data, where the complete sentences are helpful for grounding the more complex queries, including multi-term queries (e.g.*'instrumental action movie soundtrack'*), compositional queries (e.g. *'classical music with middle eastern influence'*), and even queries with negation (e.g. *'hard rock without vocals'*). Again, we find that training is surprisingly robust to annotation noise, achieving similar performance using unfiltered training text.

4.2.3 Text Triplet Classification

Table 6 lists triplet classification accuracy on evaluations introduced in Section 4.1.4. We compare MuLan text embedding against the following baselines: Sentence Transformer [46], SimCSE [45], Universal Sentence Embedding [47], and the average token embedding of BERT-base-uncased (this outperforms the `[CLS]` encoding by a large margin). All baselines are Transformer-based models with similar size to ours. The first three were trained with sentence-level contrastive loss, while BERT is trained with masked language prediction. We warmstart the MuLan text encoder using this same BERT baseline, but it is subsequently only trained with the cross-modal loss. We find that when including our long-form text annotations, the resulting text embedding model, which is now specialized to the music domain, outperforms the generic sentence embedding models. While it is not surprising in-domain text is helpful, it is remarkable that successful specialization is accomplished without using any text-only fine-tuning loss.

5. CONCLUSIONS

We presented a music audio and natural language joint embedding model trained with an unprecedented scale of weakly paired text and audio data. Our experiments demonstrate the versatility of the natural language interface in a range of applications. The pretrained audio embeddings also achieve SOTA transfer learning performance on music tagging benchmarks. This is a first attempt at building a free-form natural language interface for music audio and there is plenty of room for improvement. Specifically, we believe improved text filtering methods that better distinguish weak signal from absolute noise will result in better handling of rare and subtle language constructs.

6. REFERENCES

[1] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. Le, Y.-H. Sung, Z. Li, and T. Duerig, "Scaling up visual and vision-language representation learning with noisy text supervision," in *Proc. of the International Conference on Machine Learning (ICML)*, 2021.

[2] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. of the International Conference on Machine Learning (ICML)*, 2021.

[3] A. S. Koepke, A.-M. Oncescu, J. Henriques, Z. Akata, and S. Albanie, "Audio retrieval with natural language queries: A benchmark study," *IEEE Transactions on Multimedia*, 2022.

[4] K. Kilgour, B. Gfeller, Q. Huang, A. Jansen, S. Wisdom, and M. Tagliasacchi, "Text-driven separation of arbitrary sounds," *arXiv:2204.05738*, 2022.

[5] A. Nagrani, P. H. Seo, B. Seybold, A. Hauth, S. Manen, C. Sun, and C. Schmid, "Learning audio-video modalities from image captions," *arXiv:2204.00679*, 2022.

[6] K. Drossos, S. Lipping, and T. Virtanen, "Clotho: An audio captioning dataset," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[7] C. D. Kim, B. Kim, H. Lee, and G. Kim, "AudioCaps: Generating captions for audios in the wild," in *Proc. of the Conference of the North American Chapter of the Association for Computational Linguistics (NAACL): Human Language Technologies*, 2019.

[8] X. Favory, K. Drossos, T. Virtanen, and X. Serra, "COALA: Co-aligned autoencoders for learning semantically enriched audio representations," *arXiv:2006.08386*, 2020.

[9] ——, "Learning contextual tag embeddings for cross-modal alignment of audio and tags," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[10] Y. Gong, Y.-A. Chung, and J. Glass, "AST: Audio spectrogram transformer," *arXiv:2104.01778*, 2021.

[11] Y. Gong, C.-I. J. Lai, Y.-A. Chung, and J. Glass, "SSAST: Self-supervised audio spectrogram transformer," *arXiv:2110.09784*, 2021.

[12] P. Hamel, M. E. Davies, K. Yoshii, and M. Goto, "Transfer learning in MIR: Sharing learned latent representations for music audio classification and similarity," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2013.

[13] A. Van Den Oord, S. Dieleman, and B. Schrauwen, "Transfer learning by supervised pre-training for audio-based music classification," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2014.

[14] K. Choi, G. Fazekas, M. Sandler, and K. Cho, "Transfer learning for music classification and regression tasks," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2017.

[15] P. Dhariwal, H. Jun, C. Payne, J. W. Kim, A. Radford, and I. Sutskever, "Jukebox: A generative model for music," *arXiv:2005.00341*, 2020.

[16] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[17] J. F. Gemmeke, D. P. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "AudioSet: An ontology and human-labeled dataset for audio events," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[18] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2011.

[19] A. Jansen, M. Plakal, R. Pandya, D. P. Ellis, S. Hershey, J. Liu, R. C. Moore, and R. A. Saurous, "Unsupervised learning of semantic audio representations," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018.

[20] A. Saeed, D. Grangier, and N. Zeghidour, "Contrastive learning of general-purpose audio representations," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[21] N. Turpault, R. Serizel, and E. Vincent, "Semi-supervised triplet loss based learning of ambient audio embeddings," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2019.

[22] J. Spijkervet and J. A. Burgoyne, "Contrastive learning of musical representations," *arXiv:2103.09410*, 2021.

[23] R. Castellon, C. Donahue, and P. Liang, "Codified audio language modeling learns useful representations for music information retrieval," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2021.

[24] S. Oramas, O. Nieto, F. Barbieri, and X. Serra, "Multi-label music genre classification from audio, text, and images using deep features," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2017.

[25] Q. Huang, A. Jansen, L. Zhang, D. P. Ellis, R. A. Saurous, and J. Anderson, "Large-scale weakly-supervised content embeddings for music recommendation and tagging," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[26] A. Ferraro, X. Favory, K. Drossos, Y. Kim, and D. Bogdanov, "Enriched music representations with multiple cross-modal contrastive learning," *IEEE Signal Processing Letters*, vol. 28, pp. 733–737, 2021.

[27] A. Guzhov, F. Raue, J. Hees, and A. Dengel, "Audioclip: Extending CLIP to image, text and audio," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[28] F. Font, G. Roma, and X. Serra, "Freesound technical demo," in *Proc. of the ACM International conference on Multimedia (MM)*, 2013.

[29] H. Xie and T. Virtanen, "Zero-shot audio classification via semantic embeddings," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 1233–1242, 2021.

[30] K. J. Piczak, "ESC: Dataset for environmental sound classification," in *Proc. of the ACM International Conference on Multimedia (MM)*, 2015.

[31] M. Won, J. Salamon, N. J. Bryan, G. J. Mysore, and X. Serra, "Emotion embedding spaces for matching music to stories," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2021.

[32] I. Manco, E. Benetos, E. Quinton, and G. Fazekas, "Learning music audio representations via weak language supervision," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[33] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. of European Conference on Computer Vision (ECCV)*, 2020.

[34] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv:1807.03748*, 2018.

[35] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *Proc. of International Conference on Machine Learning (ICML)*, 2020.

[36] H. Lee, J. Lee, J. Y.-H. Ng, and P. Natsev, "Large scale video representation learning via relational graph clustering," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[37] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold *et al.*, "CNN architectures for large-scale audio classification," in *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017.

[38] M.-W. C. Kenton, Jacob Devlin and L. K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL-HLT*, 2019, pp. 4171–4186.

[39] TensorflowHub, "https://tfhub.dev/tensorflow/bert_en_uncased_L-12_H-768_A-12/4," 2022.

[40] E. Law and L. Von Ahn, "Input-agreement: a new mechanism for collecting data using human computation games," in *Proc. of the ACM SIGCHI Conference on Human Factors in Computing Systems*, 2009.

[41] R. Typke, F. Wiering, R. C. Veltkamp, J. D. Reiss, G. A. Wiggins *et al.*, "A survey of music information retrieval systems," in *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2005.

[42] M. Won, A. Ferraro, D. Bogdanov, and X. Serra, "Evaluation of cnn-based automatic music tagging models," *arXiv:2006.00751*, 2020.

[43] A. Ettinger, "What BERT is not: Lessons from a new suite of psycholinguistic diagnostics for language models," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 34–48, 2020.

[44] G. N. C. Tejada, J. Scholtes, and G. Spanakis, "A study of BERT's processing of negations to determine sentiment," in *Proc. of the Benelux Conference on Artificial Intelligence and Belgian-Dutch Conference on Machine Learning*, 2021.

[45] T. Gao, X. Yao, and D. Chen, "SimCSE: Simple contrastive learning of sentence embeddings," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021, pp. 6894–6910.

[46] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using siamese bert-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[47] D. Cer, Y. Yang, S.-y. Kong, N. Hua, N. Limtiaco, R. S. John, N. Constant, M. Guajardo-Cespedes, S. Yuan, C. Tar *et al.*, "Universal sentence encoder," *arXiv:1803.11175*, 2018.

MELOFORM: GENERATING MELODY WITH MUSICAL FORM BASED ON EXPERT SYSTEMS AND NEURAL NETWORKS

Peiling Lu[1] **Xu Tan**[*1] **Botao Yu**[2]
Tao Qin[1] **Sheng Zhao**[3] **Tie-Yan Liu**[1]
[1] Microsoft Research Asia, Beijing, China
[2] Nanjing University, Nanjing, China
[3] Microsoft Azure Speech, Beijing, China
{peil,xuta,taoqin,szhao,tyliu}@microsoft.com, btyu@smail.nju.edu.cn

ABSTRACT

Human usually composes music by organizing elements according to the musical form to express music ideas. However, for neural network-based music generation, it is difficult to do so due to the lack of labelled data on musical form. In this paper, we develop MeloForm, a system that generates melody with musical form using expert systems and neural networks. Specifically, 1) we design an expert system to generate a melody by developing musical elements from motifs to phrases then to sections with repetitions and variations according to pre-given musical form; 2) considering the generated melody is lack of musical richness, we design a Transformer based refinement model to improve the melody without changing its musical form. MeloForm enjoys the advantages of precise musical form control by expert systems and musical richness learning via neural models. Both subjective and objective experimental evaluations demonstrate that MeloForm generates melodies with precise musical form control with 97.79% accuracy, and outperforms baseline systems in terms of subjective evaluation score by 0.75, 0.50, 0.86 and 0.89 in structure, thematic, richness and overall quality, without any labelled musical form data. Besides, MeloForm can support various kinds of forms, such as verse and chorus form, rondo form, variational form, sonata form, etc. Music samples generated by MeloForm are available via this link [1], and our code is available via this link [2].

1. INTRODUCTION

Melody is often composed of hierarchical motifs, phrases and sections with repetitions and variations given musical form [1]. This organized structure provided by musical form can help better express music ideas. For example, verse and chorus form is widely used in popular music. The repetition of verse and chorus sections helps emphasize music ideas, while the contrast between verse and chorus can create more emotional intensity. Automatic melody generation with pre-given musical form based on purely data driven technology is difficult due to the lack of labelled data on musical form. Previous work attempt to generate melody with structure information, but still suffers from the following issues: 1) They generate melodies with repetitive patterns implicitly either by learning long-term dependency [2–4] or representing the repetition structure with same harmony, rhythm patterns, etc [5–7]. However, repetitive patterns are far from the exact musical form. 2) They generate melodies with bar-level structure explicitly by learning the relationship between bars [8, 9], but bar-level structure is still not the exact musical form. 3) They generate melodies with repetitive phrases and sections either by detecting phrase labels with rules and algorithms [10] or using human labelling [11]. But it is hard for rules or algorithms to detect precise musical form, and it costs much for hiring people to label this musical form data. Furthermore, all of these work aim to generate melodies with some kind of repetitive patterns, but musical form is constructed to express music ideas by developing the hierarchical structural units (i.e., motifs, phrases and sections) with repetitions and variations. Simply repeating some fragments, or even phrases and sections, without considering the relationships among these hierarchical structures, is superficial.

Although it is difficult to collect labeled musical form data through detection algorithm or human labeling, it is much easier for expert systems to generate melodies with musical form. However, experts systems may suffer from monotonous musicality because of handcraft rules. On the other hand, neural networks are capable of creating melodies with rich expressions by learning the data distribution, but it is hard to precisely control the musical form. Considering the complementary characteristics of these two systems, we come up with a method that can leverage the advantages and make up for the shortcomings.

In this paper, we develop MeloForm, a system that generates melody with musical form using expert systems and neural networks. The expert system is designed to generate

* Corresponding author: Xu Tan, xuta@microsoft.com

[1] https://ai-muzic.github.io/meloform/

[2] https://github.com/microsoft/muzic/tree/main/meloform

synthetic melodies with precise musical form. It develops the motifs to phrases then to sections, which are arranged by repetitions and variations according to the pre-given musical form. The encoder-attention-decoder Transformer based neural network is introduced to refine melodies generated by expert systems. To improve musical richness without changing musical form, we propose the refinement strategy in phrase level, the conditioning on rhythm and harmony, and the methods for differentiating sections.

MeloForm enjoys the advantages of expert systems and neural models and avoids their limitations as following: 1) Comparing with expert systems, we can generate melodies with better musical richness. 2) Comparing with the models that implicitly learn the repetitive patterns, we can generate melodies explicitly with precise musical form control. 3) Comparing with the models that depend on bar-level structures, we construct higher-level phrases and sections structures. 4) Comparing with the models using detected phrases labels or human labels, we have the labeled musical form data naturally from expert systems with zero cost and precise accuracy.

The main contributions of this work are as follows:

- We develop MeloForm, a system that generates melody with musical form using expert systems and neural networks. This system combines the best of white-box expert system and black-box neural networks for generating melodies with precise musical form and rich melodic expression without any labeled data.
- Experimental results demonstrate that MeloForm achieves precise musical form control with 97.79% accuracy without any labeled data, and outperforms baseline systems by 0.75, 0.50, 0.86 and 0.89 averagely in structure, thematic, richness and overall quality in subjective evaluation.
- MeloForm can generate melodies with various kinds of forms, such as verse and chorus form, rondo form, variational form, sonata form, etc.

2. RELATED WORK

Automatic melody generation evolves from grammar or statistical based generation [12–17] to deep learning empowered generation [2, 3, 6, 8–10, 18–31]. In this section, we introduce existing neural networks and expert systems for melody generation with musical structure.

2.1 Neural Networks for Melody Generation

Generating structured melody has attracted more attention when modeling long music sequence. Previous work address this problem as following: 1) They implicitly learn the long-term dependency or represent repetitive structure with same musical elements to generate melodies with repetitive patterns. Music Transformer [3] introduces a relative attention mechanism to capture long-term dependency. Theme Transformer [4] proposes a novel gated parallel attention module for generation with theme-based conditioning. Another work [2] presents a hierarchical recurrent neural network to model the note-beat-bar structure. Other methods [5–7] condition the model with same musical features (e.g., harmony and rhythm patterns) to represent the repetitive structure. 2) They explicitly model the bar-level structure for generating melodies one bar after another. In [9, 32], the authors leverage the bar related self-similarity matrix to model the relationship between bars for guiding the melody generation. MELONS [8] constructs a bar-level structure graph for generating melodies with clear bar-level structures. 3) They collect labeled musical data by detection algorithms for generating repetitions for melodies. Repetitive patterns from melodies are detected by music analysis algorithms in [33], while the boundaries of repetitive phrases are recognized in [10]. All of these works can help realize repetitive patterns for melody in some degree, but they still cannot model precise musical form.

2.2 Expert Systems for Melody Generation

Back to 18th century, a system called music dice game is developed for randomly generating music from precomposed options [34]. Recently, much work still investigate rule-based algorithm compositions for specific purpose. The author in [35] comes up with a rule-based algorithm to generate melody note sequence, which is constructed for comparison with the machine learning based compositions. dMelodies [36] combines the designed latent factors to create 2-bar melodies for improving data diversity in disentanglement learning. However, none of them consider the rules for generating melodies with musical form. Computoser [37] proposes a hybrid probability/rule based algorithm for music composition, which based on rules about structure, rhythm, repetition, variations, endings, etc. But it did not provide the methods about how to arrange these elements with musical form. And the method for constructing a phrase by multiple different motifs may brings about divergence from music ideas. In [38], repetition of phrases is also considered at a rhythmical level and concerning pitch intervals, but the melody in each phrase is generated one note at a time, which is different from a common composition process by developing a phrase from a motif. Comparing with these systems, the expert system in MeloForm generates melodies by considering the musical form as the hierarchical structure of motifs, phrases and sections with repetitions and variations, which can better express music ideas.

3. METHOD

To combine the advantages of precise musical form control by expert systems and musical richness learning by neural networks, we develop MeloForm that is shown in Figure 2(a), which contains two modules: 1) expert systems for generating synthetic melodies with musical forms; 2) Transformer based neural networks for refining the generated melodies from expert systems.