

# A Pairwise Reward Model on Frozen Music Encoders for MIREX 2026 Music Evaluation via CMI-RewardBench

Hsieh-Yu Yang  
National Central University

Takato Horii  
Osaka University

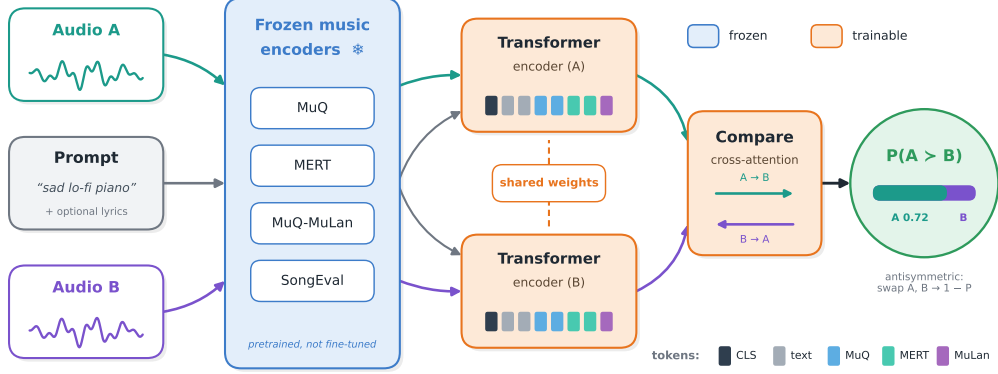


Figure 1: The system overview. Frozen encoders extract features from each clip, and the Siamese Transformer and cross-attention comparison produce  $P(A \succ B)$ .

## ABSTRACT

We present a pairwise reward model for the MIREX 2026 Music Evaluation via CMI-RewardBench task, which predicts listener preferences between two generated music clips given a prompt. We use frozen features from MuQ, MERT, MuQ-MuLan, and SongEval as input to a lightweight Siamese Transformer. The model jointly compares the two clips using an antisymmetric scoring function, ensuring that its prediction is invariant to their input order. We train on public Music Arena votes together with CMI-Pref, AIME, and CMI-Pref-Pseudo. Our ensemble reached 79.1% accuracy; CMI-RM reached 73.1% on the 2025 Music Arena test split. On held-out 2026 battles, a model trained on January–May votes reached 85.6% accuracy, while CMI-RM reached 75.6%. Results on CMI-Pref are comparable to those of the baseline.

## 1. INTRODUCTION

Given a text prompt, which may include lyrics, the task is to predict which of two clips, A or B, listeners prefer. Systems are scored by accuracy on recent crowd-sourced battles collected in the Music Arena [1] format. The baseline, CMI-RM [2], is built on frozen MuQ-MuLan embeddings. It is first pre-trained on about 110k comparisons labelled by an audio language model, then fine-tuned on human annotations. Like CMI-RM, we keep the encoders frozen. What we change is that we use several encoders instead of one, compare the two clips jointly, and train on in-domain crowd votes.

## 2. SYSTEM

### 2.1 Features

Audio is resampled to 24kHz mono and split into 30 s chunks. We use at most 240 s of each clip. From MuQ [3] we keep the output of every hidden layer, and from MERT-

330M [4] the output of every second layer. In both cases the frame-level outputs are averaged within one-second windows. MuQ-MuLan embeds the audio in 10 s segments, while the SongEval predictor [5] outputs five scores for the whole clip. The prompt and lyrics are encoded with the MuQ-MuLan text encoder. All encoders are kept frozen.

### 2.2 Model

A learnable softmax over layers summarises MuQ and MERT. All inputs are projected to 256 dimensions and arranged as one token sequence per clip ([CLS], prompt, lyrics, SongEval, MuQ, MERT and MuQ-MuLan tokens), which a two-layer Transformer with shared weights encodes. A pointwise head scores each clip,  $s(\cdot)$ , and a comparative head scores each clip after cross-attending to the other,  $c(\cdot | \cdot)$ . The pair logit

$$\ell(A, B) = [s(A) - s(B)] + [c(A|B) - c(B|A)] \quad (1)$$

satisfies  $\ell(A, B) = -\ell(B, A)$ , so the prediction does not depend on the order of the clips. Three output heads are trained with a Bradley–Terry loss [6]: overall preference (Music Arena), musicality and prompt alignment (CMI-Pref, AIME). The submission uses the overall head.

### 2.3 Encoder fusion

Concatenating the frame-level MuQ and MERT features did not work well in our early experiments. On held-out 2026 votes it was worse than using either encoder alone, probably because we only have a few thousand crowd votes to train on. Averaging the MuQ and MERT frames over 10 s windows and adding modality dropout [7] helped. During training, each encoder stream is dropped with probability 0.3, and both clips of a pair lose the same streams. Accuracy on held-out 2026 votes went from 82.0% to 85.6%.

## 2.4 Submitted ensemble

The submitted system is an ensemble of three model variants, each trained with five random seeds. The first is the model described above. The second is the same model, but pre-trained for two epochs on CMI-Pref-Pseudo before it sees any human votes. The third uses only MuQ-MuLan features. We average the logits of all 15 models.

## 3. TRAINING DATA

Our Music Arena training data are the public releases from January to July 2026, 2,320 decisive votes in total. We never train on the July to December 2025 battles, which we use only for testing. We also use the CMI-Pref [2] training split, which has expert votes on musicality and alignment, and 13,978 of the 15,600 human votes in AIME [8], all on 10 s excerpts. The remaining AIME votes are held out. We also use CMI-Pref-Pseudo, whose labels come from Qwen3-Omni [9], after removing comparisons that share a prompt with the CMI-Pref test set. We did not use the hidden test set, the live Music Arena platform, or generator names as input.

## 4. RESULTS

We report two versions of our system. To measure accuracy on recent battles, which are closest to the test set, we trained a development model on January–May 2026 only and held out June–July 2026 (660 votes). The submitted ensemble is then retrained with June–July included, because these are the most recent votes available; as a consequence it can only be evaluated on the official 2025 split (1,340 votes) and on CMI-Pref. Table 1 compares both with CMI-RM, run with the official code and weights. The gain of the submitted ensemble over CMI-RM on 2025 is significant (McNemar test,  $p = 1.3 \cdot 10^{-6}$ ).

Table 1: Accuracy (%). Dev.: one configuration (Section 2.3), three seeds, trained on January–May 2026. Submitted: the 15-model ensemble of Section 2.4, trained on January–July 2026.

| System            | Music Arena |      | CMI-Pref |      |
|-------------------|-------------|------|----------|------|
|                   | 2026        | 2025 | mus      | ali  |
| CMI-RM (baseline) | 75.6        | 73.1 | 78.0     | 73.4 |
| Ours, dev.        | 85.6        | 79.6 | 78.4     | 73.2 |
| Ours, submitted   | n/a         | 79.1 | 79.4     | 73.2 |

## 5. IMPLEMENTATION

The trainable part has 2.8 M parameters. On cached features, training one model takes 5–20 minutes on one NVIDIA RTX 5090. Inference with the 15-model ensemble takes about one second per pair. The submission package contains the inference code (python main.py --path annotations.jsonl) and all model weights.

## REFERENCES

[1] Y. Kim, W. Chi, A. N. Angelopoulos, W.-L. Chiang, K. Saito, S. Watanabe, Y. Mitsufuji, and C. Donahue, “Music Arena: Live evaluation for text-to-music,” in *Advances in Neural Information Processing Systems 38 (NeurIPS 2025), Creative AI Track*, 2025.

[2] Y. Ma, H. Xia, H. Gao, W. Chen, Y. Ye, Y. Yang, S. Chang, M. Ding, Y. Li, R. Yuan, S. Dixon, and E. Benetos, “CMI-RewardBench: Evaluating music reward models with compositional multimodal instruction,” in *International Conference on Machine Learning (ICML)*, 2026.

[3] H. Zhu, Y. Zhou, H. Chen, J. Yu, Z. Ma, R. Gu, Y. Luo, W. Tan, and X. Chen, “MuQ: Self-supervised music representation learning with mel residual vector quantization,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 33, pp. 3653–3664, 2025.

[4] Y. Li, R. Yuan, G. Zhang, Y. Ma, X. Chen, H. Yin *et al.*, “MERT: Acoustic music understanding model with large-scale self-supervised training,” in *International Conference on Learning Representations (ICLR)*, 2024.

[5] J. Yao, G. Ma, H. Xue, H. Chen, C. Hao, Y. Jiang, H. Liu, R. Yuan, J. Xu, W. Xue, H. Liu, and L. Xie, “SongEval: A benchmark dataset for song aesthetics evaluation,” *arXiv preprint arXiv:2505.10793*, 2025.

[6] R. A. Bradley and M. E. Terry, “Rank analysis of incomplete block designs: I. The method of paired comparisons,” *Biometrika*, vol. 39, no. 3/4, pp. 324–345, 1952.

[7] N. Neverova, C. Wolf, G. Taylor, and F. Nebout, “ModDrop: Adaptive multi-modal gesture recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1692–1706, 2016.

[8] F. Grötschla, A. Solak, L. A. Lanzendörfer, and R. Wattenhofer, “Benchmarking music generation models and metrics via human preference studies,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2025.

[9] J. Xu, Z. Guo, H. Hu, Y. Chu, X. Wang, J. He *et al.*, “Qwen3-Omni technical report,” *arXiv preprint arXiv:2509.17765*, 2025.