Dual Time Machines: Interpreting Temporal Circuits in Symbolic Music and Natural Language

Yutong He, Shaowei Zhang

Keywords: Temporal Relation Classification, Multi-modality, Transformer, Attention Mechanism, Mechanistic Interpretability.

Abstract. With the hypothesis that a dual-branch Transformer might reveal both modality-specific and shared neural mechanisms for temporal reasoning across language and symbolic music, a MusicBERT–RoBERTa model is leveraged to predict temporal relations of language and music corpus. The results show that temporal attention concentrates in mid-layer heads for text but in early-layer heads for music, with only one weakly influential neuron common to both branches. These findings suggest that, under current setup, contrary to the hypothesis, temporal reasoning circuits are basically modality-specific, motivating richer music pre-training and deeper causal probes to confirm.

1 Introduction

Cognitive science suggests that language and music share neural mechanisms for temporal reasoning, involving regions such as the Broca area and the basal ganglia [1, 2]. However, this theory lacks validation through artificial intelligence.

From the perspective of mechanism interpretability, we raise the question: Are there specific behaviors of Transformer-based models when encoding language and music input? If so, can AI models demonstrate similar shared mechanisms? Specifically, do certain neurons or attention patterns specialize in temporal reasoning in both modalities?

This project aims to investigate whether shared mechanisms for temporal reasoning exist in AI models by analyzing embeddings and attention weights in a dual-branch BERT model.

2 Research Hypothesis

The behaviors of BERT-based models can be interpreted when completing the task of temporal reasoning on both natural language and music.

Intuitively, aligning with the principles of neuroscience, Specific neurons or attention patterns will emerge in the model to handle temporal reasoning tasks and show consistent activation across modalities.

3 Data

3.1 *Dataset*

3.1.1 Language Temporal Reasoning Dataset

For language dataset, we use **TRAM-Benchmark[3]** dataset.

TRAM is a benchmark covering ten temporal reasoning tasks formatted as multiple-choice questions. Each question has one correct answer to ensure clarity. The dataset draws from existing NLU datasets, human-written templates, web sources, and generated programs, with answers produced through expert annotation and automated methods.

Since our task merely focuses on temporal relation classification, we only adopt the relation dataset among all. According to Figure 8, there are around one million data samples in the dataset, all of which are derived from the TempEval-3[4] corpus.

Instead of expecting answers generated from 3-way MC (Multiple Choices) by a Large Language Model, we train a linear classifier as the last layer to complete the task.

Among the dataset, data samples are annotated with temporal relations like "before" and "after.", as shown in Table 1.

Table 1: **TRAM examples**

.

Question	Answer	Label Ratio
The central Bank Negara Tuesday imposed new lending limits on shares and luxury properties to thwart excessive market speculation and quell soaring property prices. What is the relationship between the event 'imposed' and the event 'quell'?	AFTER	35%
Rules for lending against stocks and unit trusts were also redefined. KUALA LUMPUR, April 1 , 1997 (AFP). What is the relationship between the event 'redefined' and the time 'April 1 , 1997'?	BEFORE	39%
The Royal Trophy, to be held on January 5-8 at Thailand's world-class Amata Spring Country Club, has been sanctioned by the Asian, Japan and European Tours. What is the relationship between the event 'held' and the time 'January 5-8'?	IS_INCLUDED	16%
The final report of the 10-month Cole inquiry, an investigation by former judge Terence Cole, concluded that AWB knowingly paid bribes to Baghdad to win wheat contracts and then misled the Australian government and the United Nations. What is the relationship between the event 'concluded' and the event 'paid'?	SIMULTANEOUS	10%

3.1.2 Music Temporal Reasoning Dataset

For the music dataset, we adopt the Clean MIDI subset from the **Lakh MIDI Dataset** [5, 6]. This subset comprises 17,256 unique MIDI files, each with filenames indicating the artist and title. It spans a wide range of musical styles, providing diverse and representative coverage across genres.

The MIDI (Musical Instrument Digital Interface) format is a symbolic, event-based representation of music. Each note is encoded as a discrete event with attributes such as pitch, duration, velocity, and instrument, without containing any actual audio.

We then automatically extract pairs of music clips (each lasting 15–30 seconds) at random from the 17,256 unique MIDI files, with temporal relations such as "Before" and "Simultaneous" between them.

For clarity and consistency, we ensure that the music samples are of the same size as the text samples, and that the labels are balanced across the music data.

3.2 *Task Design*

Both the language and music tasks involve predicting temporal relations between two sequences.

Since Allen's original 13-way classification is often too fine-grained for linear classifiers, we simplify the task to a 4-way classification: `['BEFORE', 'AFTER', 'IS_INCLUDED', 'SIMULTANEOUS']`. This setup is applied consistently to both language and music domains to enable a more balanced and comparable evaluation.

4 Methodology

4.1 *Model Architecture*

To investigate how attention weights contribute to temporal reasoning across music and language, we design a dual-branch Transformer model, as illustrated in Figure 6. Inputs from each modality are independently encoded by their respective branches to generate modality-specific embeddings.

For the music branch, we employ MusicBERT [7], a Transformer-based model pretrained specifically on symbolic music representations. MusicBERT introduces an encoding method called OctupleMIDI, which explicitly encodes every note in a MIDI segment into an octuple token consisting of eight attributes: **Time Signal**, **BPM**, **Bar**, **Position**, **Instrument**, **Pitch**, **Duration**, and **Velocity** (see Figure 7). We use the pretrained checkpoints officially provided in their GitHub repository.

Since MusicBERT is based on the RoBERTa architecture [8], we adopt a pretrained RoBERTa model to extract textual features, ensuring a fair and consistent comparison. The pretrained RoBERTa checkpoints are obtained from Hugging Face.

The embeddings from both branches are subsequently passed into a shared embedding module comprising four Transformer encoder blocks. Finally, two classifiers separately predict the relation labels corresponding to pairs of music segments or text sentences. The more detailed description of the architecture is attached in Appendix.

5 Results

5.1 *Training*

The accuracy trends on the training, validation, and test sets throughout the training process are shown in Figure 1. The best test accuracy reaches approximately 44% for the music task and 84% for the language task, while training accuracy approaches 99% for both. This indicates clear overfitting in the music task.

The weaker performance on the music task may be attributed to limitations in feature extraction by the pretrained MusicBERT model, as pretrained language models are generally more mature and effective, and it is more complicated to understand music tokens for Transformer-like models.

We save the checkpoints from every epoch for further analysis.

5.2 *Attention Analysis*

With the checkpoints trained on both datasets, we next look inside the shared Transformer blocks to check whether the attention weights focus on particular tokens across the two modalities.

For text data, we follow the pipeline below:

- We randomly choose 250 samples from the test set in dataset for each temporal relation

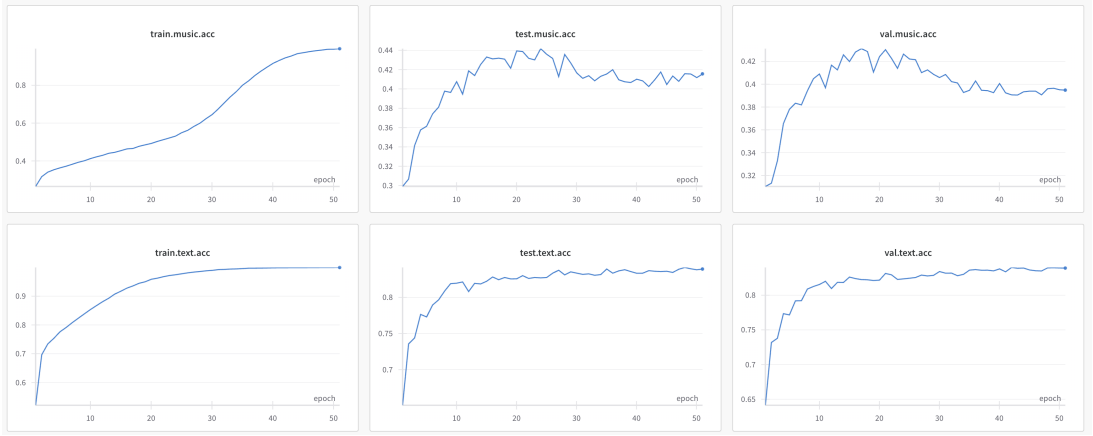


Figure 1: **Accuracy**.

label.

- We reload the checkpoints for every epoch, to understand the trends in attention behaviors.
- We locate the target tokens in each sample. According to Table 1 Every text sample follows a prompted question ("What is the relationship between the event/time '**a**' and the event/time '**b**'?"). We manage to find all relative indices of token from **a** and **b**.
- With the index lists, we then extract the average attention score head-wisely between the event and time in each text sample.

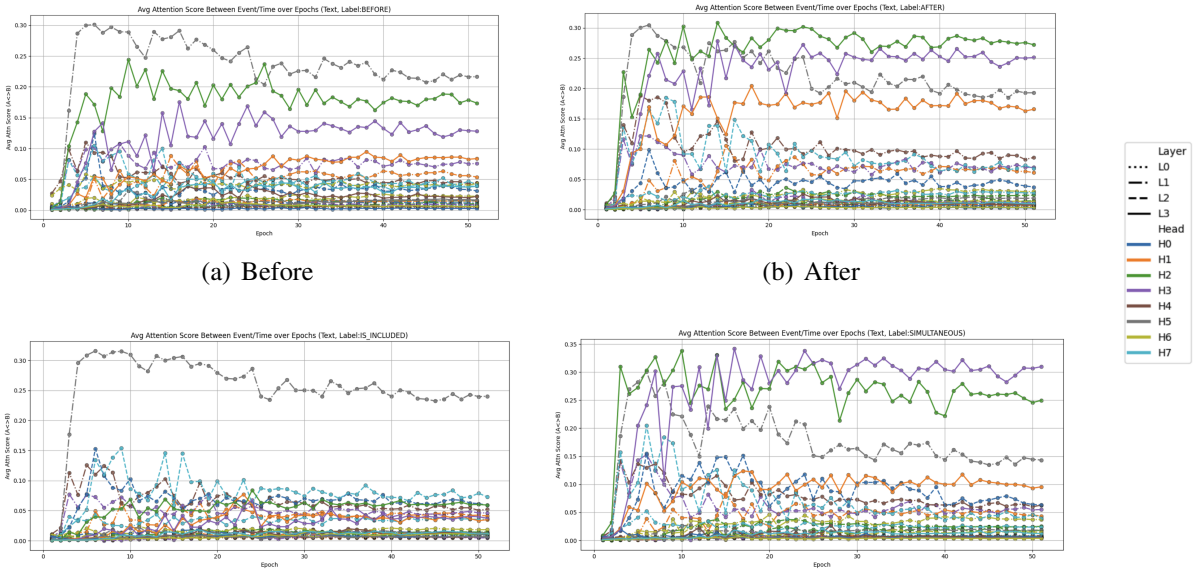
According to Figure 2 and Table 2, our analysis reveals that attention to event–time token pairs is consistently concentrated within a small subset of Transformer heads. Specifically, four heads—layer 1, head 5 (**L1-H5**), and layer 3, heads 1, 2, and 3 (**L3-H1**, **L3-H2**, and **L3-H3**)—appear repeatedly among the top-five strongest heads for all temporal relations tested (BEFORE, AFTER, IS_INCLUDED, SIMULTANEOUS). A single additional head varies by relation type, indicating minor, relation-specific specialization beyond this stable core.

Table 2: **[Text] Top-5 attention heads for each temporal relation label over all epochs**. Heads are denoted by Layer-Head indices.

Label	**Top-5 Attention Heads (Layer-Head)**
BEFORE	1-5, 3-2, 3-3, 1-3, 3-1
AFTER	3-2, 3-3, 1-5, 3-1, 1-4
IS_INCLUDED	1-5, 2-7, 2-0, 1-4, 3-2
SIMULTANEOUS	3-3, 3-2, 1-5, 3-1, 2-0
ALL (Aggregate)	1-5, 3-2, 3-3, 3-1, 1-4

For music data, we apply a similar pipeline:

- We randomly sample 250 examples per temporal relation label from the music test set.
- We reload checkpoints from each epoch to analyze the evolution of attention behaviors.
- Unlike the text data scenario, the two events here correspond to entire sequences of OctupleMIDI tokens. Because embeddings from two music segments are concatenated



(a) Before (b) After

(c) Is_included (d) Simultaneous

Figure 2: **Attention Analysis for Text per Label**

before the shared Transformer modules, attention matrices split naturally into diagonal blocks (within-segment attention) and off-diagonal blocks (cross-segment attention).

- To quantify how attention connects these two sequences, we compute a symmetric attention measure by summing average attention scores from segment A to B and from segment B to A.

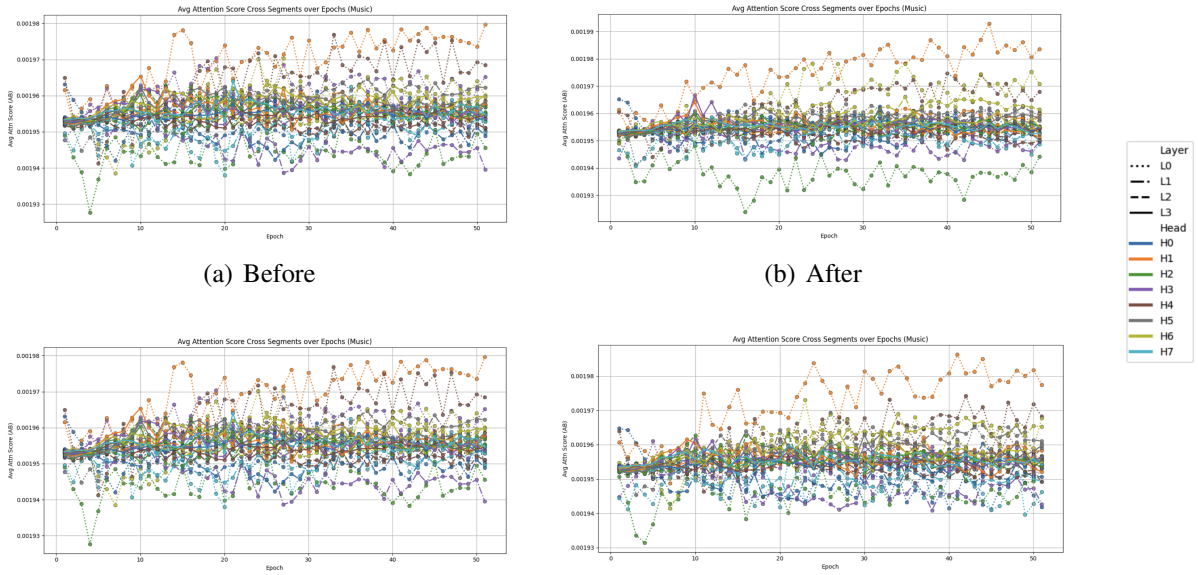
In the music modality, attention is similarly concentrated in a small set of heads, with layer 0 head 1 (**L0-H1**) and head 4 (**L0-H4**) consistently appearing across all labels. Heads **L0-H6**, **L1-H6**, and **L1-H5** also recur frequently, suggesting shared temporal-attention patterns. Unlike text, where key heads emerge in mid-layers, music relies more on early-layer attention, indicating that temporal cues in music may be captured at lower-level representations. These results show modality-specific specialization in attention behavior.

Table 3: **[Music] Top-5 attention heads for each temporal relation label over all epochs**. Heads are denoted by Layer-Head indices. Results are aggregated from both directions (A↔B).

Label	Top-5 Attention Heads (Layer-Head)
BEFORE	0-1, 0-4, 0-3, 1-5, 3-6
AFTER	0-1, 0-6, 0-4, 1-6, 0-5
IS_INCLUDED	0-1, 0-4, 1-6, 0-6, 0-5
SIMULTANEOUS	0-1, 0-4, 1-6, 0-5, 1-5
ALL (Aggregate)	0-1, 0-4, 0-6, 1-6, 1-5

To identify which note attributes the model relies on for temporal reasoning, we group token indices by attribute and track their mean attention weights across all training samples during training process.

As shown in Figure 4, the model first focuses more on Time Signal tokens. Over time, its attention shifts toward Position and BPM, which describe more detailed timing information. This suggests that the model gradually learns to use finer-grained timing features instead of just relying on the overall time signal.



(a) Before (b) After

(c) Is_included (d) Simultaneous

Figure 3: **Attention Analysis for Music per Label**

5.3 *Probing*

We perform activation patching and probing[9] based on the pooled features extracted from the shared Transformer blocks, with a hidden size of 768. A linear probe is trained on the representations to perform temporal relation classification. To assess neuron contribution, we apply zero patching—setting individual neuron activations to zero—and measure the drop in accuracy.

After training on both text and music data, we identify the top 20 neurons whose patching most degrades performance for each modality. Neurons common to both sets are considered candidates for shared, modality-independent temporal reasoning.

Table 4 suggests that only neuron 346 may function as a shared neuron across both modalities for temporal reasoning. However, as shown in Figure 5, focusing on neuron 346, we observe no clear trends or meaningful correlations between accuracy changes and the training process. The observed accuracy differences are minor and likely insignificant.

Table 4: **Top 20 neurons with highest accuracy drop after zero-patching**

Modality	**Top-20 Most Impactful Neurons Indices**
Music	**346**, 435, 652, 153, 605, 219, 9, 156, 577, 295, 306, 447, 678, 271, 276, 127, 164, 252, 296, 765
Text	588, 729, 574, 250, 607, 753, 458, 71, 169, 440, 581, 722, 335, **346**, 57, 338, 280, 249, 286, 240

6 Conclusion

This project examined how a dual-branch Transformer performs temporal-relation reasoning in language and music, combining performance, attention analysis, activation patching and probing. Empirically, the model attains strong accuracy on the language task (84%) but far lower performance on the music task (44%), with clear over-fitting on the latter.

Attention analysis shows that temporal reasoning is funneled through a small, modality-specific

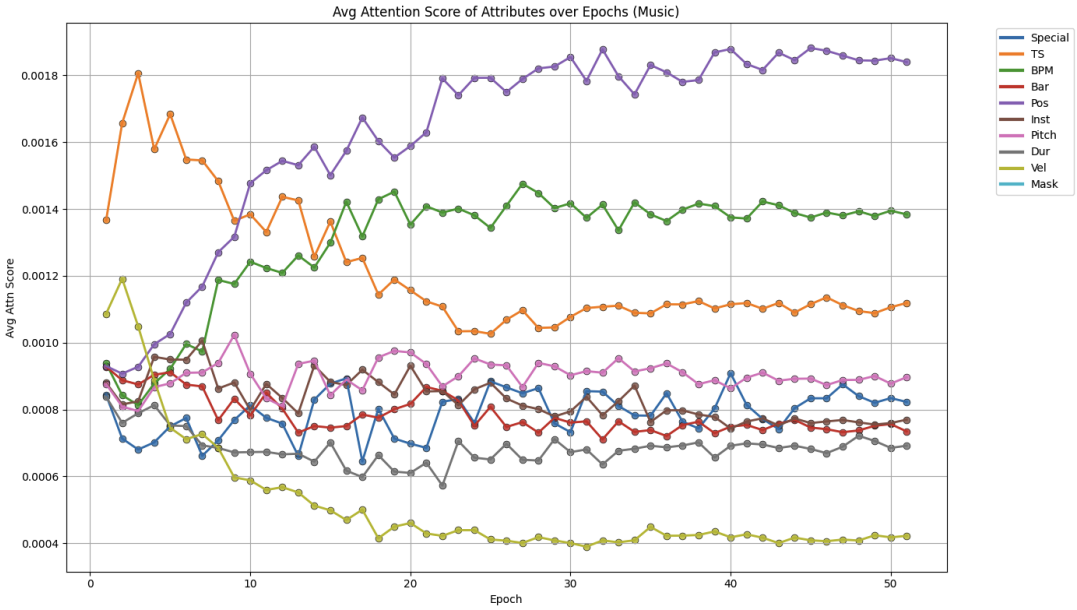
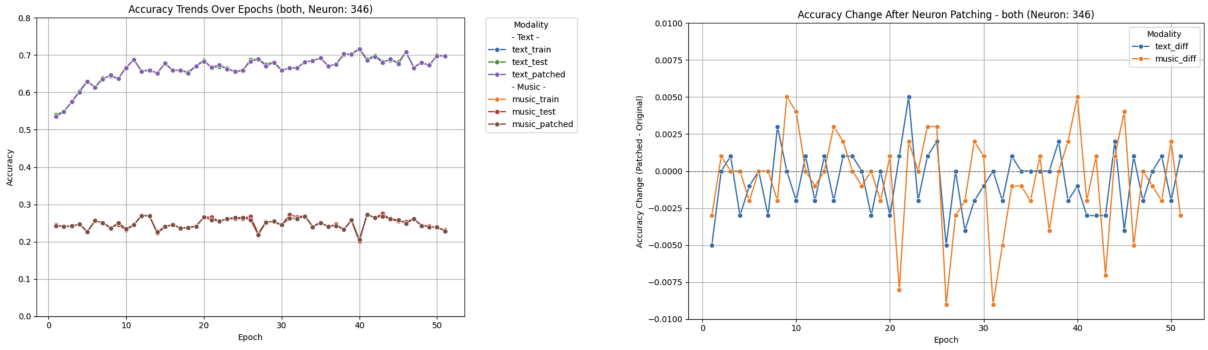


Figure 4: **Attributes Most Attended**.



(a) Accuracy (b) Accuracy Difference

Figure 5: **Accuracy trends over epochs before and after zero-patching for Neuron 346**

set of heads: mid-layer heads (L3-H1/2/3) and one early head (L1-H5) dominate for text, whereas early-layer heads (L0-H1/4/6 and L1-H6/5) dominate for music.

The model's focus on musical Time-Signal tokens gradually shifts toward finer-grained Position and BPM cues, mirroring the coarse-to-fine pattern observed in language.

Causal patching reveals many modality-critical neurons but only one (index 346) that ranks among the most influential units in both modalities; its actual influence on accuracy, however, is marginal, offering little evidence for a shared, modality-agnostic circuit.

In conclusion, our project confirms that Transformer models exhibit interpretable and time-related patterns when performing temporal reasoning, supporting the first part of our hypothesis. However, these patterns are largely modality-specific rather than truly shared. Although a small core of heads and neurons appear in both branches, their impact is modest, and most of the temporal reasoning circuitry diverges between language and music.

7 Discussion

The model we currently apply provides only weak evidence for a shared mechanism analogous to the neural overlap proposed by cognitive science. Bridging the remaining gap may require several targeted advances:

- **Better music representations.** Symbolic-music understanding is far less mature than language modelling; our results are based on a frozen MusicBERT encoder. Future work should replace it with a more powerful pretrained model, or fine-tune its upper layers to boost generalisation on temporal-reasoning tasks.
- **Longer musical context.** Memory limits forced us to cap each MIDI segment at 512 tokens, lower than MusicBERT's 8192-token capacity. Shortening clips to 3-6 seconds caused even larger accuracy drops, so we kept segments between 15 to 30 seconds. and retained the first and last n tokens while randomly sampling the middle. This inevitably discards information and likely explains the music branch's weaker performance. Therefore, efficient long-sequence mechanisms should alleviate this constraint.
- **Further causal ablations.** Our current probes operate only on pooled embeddings. Head-wise or layer-wise ablations, employed with systematic activation patching, are needed to test whether the identified attention heads truly form causal temporal-reasoning circuits.
- **Improvements on the shared architecture.** Using stacked Transformer blocks as the shared module may be too simple to encode the temporal features. Furthermore, feeding earlier embeddings from the branched encoders could better expose genuinely shared temporal representations.

These steps could clarify whether shared temporal circuits can be induced in Transformer models, rather than merely inferred from surface correlations.

Availability of data and software code

Our software code is available at the following URL:
https://github.com/WitchPuff/Time_Machines_in_Music_Text

References

[1] Aniruddh D. Patel. *Music, Language, and the Brain*. Oxford University Press, New York, 2008.

[2] Jessica A. Grahn and Matthew Brett. Rhythm and beat perception in motor areas of the brain. *Journal of Cognitive Neuroscience*, 19(5):893–906, 2007.

[3] Yuqing Wang and Yun Zhao. TRAM: Benchmarking temporal reasoning for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 6389–6415, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[4] Naushad UzZaman, Hector Llorens, Leon Derczynski, James Allen, Marc Verhagen, and James Pustejovsky. SemEval-2013 task 1: TempEval-3: Evaluating time expressions, events, and temporal relations. In Suresh Manandhar and Deniz Yuret, editors, *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 1–9, Atlanta, Georgia, USA, June 2013. Association for Computational Linguistics.

[5] Colin Raffel. *Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching*. Ph.d. dissertation, Columbia University, New York, NY, USA, 2016.

[6] Colin Raffel. The lakh midi dataset. `https://colinraffel.com/projects/lmd`, 2016. Accessed: 2025-04-15.

[7] Mingliang Zeng, Xu Tan, Rui Wang, Zeqian Ju, Tao Qin, and Tie-Yan Liu. MusicBERT: Symbolic music understanding with large-scale pre-training. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 791–800, Online, August 2021. Association for Computational Linguistics.

[8] Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. A robustly optimized BERT pre-training approach with post-training. In Sheng Li, Maosong Sun, Yang Liu, Hua Wu, Kang Liu, Wanxiang Che, Shizhu He, and Gaoqi Rao, editors, *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China, August 2021. Chinese Information Processing Society of China.

[9] Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching, 2024.

Appendix

SharedModel Architecture

```
SharedModel(
  (music_encoder): MusicEncoder(
    (musicbert): RobertaHubInterface(
      (model): MusicBERTModel(
        (encoder): MusicBERTEncoder(
          (sentence_encoder): OctupleEncoder(
            (dropout_module): FairseqDropout()
            (embed_tokens): Embedding(1237, 768, padding_idx=1)
            (embed_positions): LearnedPositionalEmbedding(8194, 768,
               padding_idx=1)
            (emb_layer_norm): LayerNorm((768,), eps=1e-05,
               elementwise_affine=True)
            (layers): ModuleList(
              (0-11): 12 x TransformerSentenceEncoderLayer(
                (dropout_module): FairseqDropout()
                (activation_dropout_module): FairseqDropout()
                (self_attn): MultiheadAttention(
                  (dropout_module): FairseqDropout()
                  (k_proj): Linear(in_features=768, out_features=768,
                     bias=True)
                  (v_proj): Linear(in_features=768, out_features=768,
                     bias=True)
                  (q_proj): Linear(in_features=768, out_features=768,
                     bias=True)
                  (out_proj): Linear(in_features=768, out_features=768,
                     bias=True)
                )
                (self_attn_layer_norm): LayerNorm((768,), eps=1e-05,
                   elementwise_affine=True)
                (fc1): Linear(in_features=768, out_features=3072, bias=True)
                (fc2): Linear(in_features=3072, out_features=768, bias=True)
                (final_layer_norm): LayerNorm((768,), eps=1e-05,
                   elementwise_affine=True)
              )
            )
            (downsampling): Sequential(
              (0): Linear(in_features=6144, out_features=768, bias=True)
            )
            (upsampling): Sequential(
              (0): Linear(in_features=768, out_features=6144, bias=True)
            )
          )
          (lm_head): RobertaLMHead(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (layer_norm): LayerNorm((768,), eps=1e-05,
               elementwise_affine=True)
          )
        )
        (classification_heads): ModuleDict(
          (topmagd_head): RobertaClassificationHead(
            (dense): Linear(in_features=768, out_features=768, bias=True)
            (dropout): Dropout(p=0.0, inplace=False)
            (out_proj): Linear(in_features=768, out_features=13, bias=True)
```

```
        )
      )
    )
  )
)
(text_encoder): TextEncoder(
  (model): RobertaForMaskedLM(
    (roberta): RobertaModel(
      (embeddings): RobertaEmbeddings(
        (word_embeddings): Embedding(50265, 768, padding_idx=1)
        (position_embeddings): Embedding(514, 768, padding_idx=1)
        (token_type_embeddings): Embedding(1, 768)
        (LayerNorm): LayerNorm((768,), eps=1e-05,
           elementwise_affine=True)
        (dropout): Dropout(p=0.1, inplace=False)
      )
      (encoder): RobertaEncoder(
        (layer): ModuleList(
          (0-11): 12 x RobertaLayer(
            (attention): RobertaAttention(
              (self): RobertaSdpaSelfAttention(
                (query): Linear(in_features=768, out_features=768,
                   bias=True)
                (key): Linear(in_features=768, out_features=768, bias=True)
                (value): Linear(in_features=768, out_features=768,
                   bias=True)
                (dropout): Dropout(p=0.1, inplace=False)
              )
              (output): RobertaSelfOutput(
                (dense): Linear(in_features=768, out_features=768,
                   bias=True)
                (LayerNorm): LayerNorm((768,), eps=1e-05,
                   elementwise_affine=True)
                (dropout): Dropout(p=0.1, inplace=False)
              )
            )
            (intermediate): RobertaIntermediate(
              (dense): Linear(in_features=768, out_features=3072,
                 bias=True)
              (intermediate_act_fn): GELUActivation()
            )
            (output): RobertaOutput(
              (dense): Linear(in_features=3072, out_features=768,
                 bias=True)
              (LayerNorm): LayerNorm((768,), eps=1e-05,
                 elementwise_affine=True)
              (dropout): Dropout(p=0.1, inplace=False)
            )
          )
        )
      )
    )
    (lm_head): RobertaLMHead(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (layer_norm): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
      (decoder): Linear(in_features=768, out_features=50265, bias=True)
    )
```

```
    )
  )
  (transformer_block): SharedTransformerBlock(
    (layers): ModuleList(
      (0-3): 4 x TransformerEncoderLayer(
        (self_attn): MultiheadAttention(
          (out_proj): NonDynamicallyQuantizableLinear(in_features=768,
              out_features=768, bias=True)
        )
        (linear1): Linear(in_features=768, out_features=2048, bias=True)
        (dropout): Dropout(p=0.1, inplace=False)
        (linear2): Linear(in_features=2048, out_features=768, bias=True)
        (norm1): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (norm2): LayerNorm((768,), eps=1e-05, elementwise_affine=True)
        (dropout1): Dropout(p=0.1, inplace=False)
        (dropout2): Dropout(p=0.1, inplace=False)
      )
    )
    (pooling): AdaptiveAvgPool1d(output_size=1)
  )
  (ffn): FeedForwardLayer(
    (ffn): Sequential(
      (0): Linear(in_features=768, out_features=3072, bias=True)
      (1): ReLU()
      (2): Linear(in_features=3072, out_features=768, bias=True)
    )
  )
  (text_classifier): TaskClassifier(
    (fc): Linear(in_features=768, out_features=4, bias=True)
  )
  (music_classifier): TaskClassifier(
    (fc): Linear(in_features=768, out_features=4, bias=True)
  )
)
```

Top-level Module	**#Params**	**Trainable**
music_encoder	103,887,074	0
text_encoder	124,697,433	0
transformer_block	22,055,936	22,055,936
ffn	4,722,432	4,722,432
text_classifier	3,076	3,076
music_classifier	3,076	3,076
Total	255,369,027	
Trainable Total		26,784,520

Table 5: **Statistics of Module Parameters**.

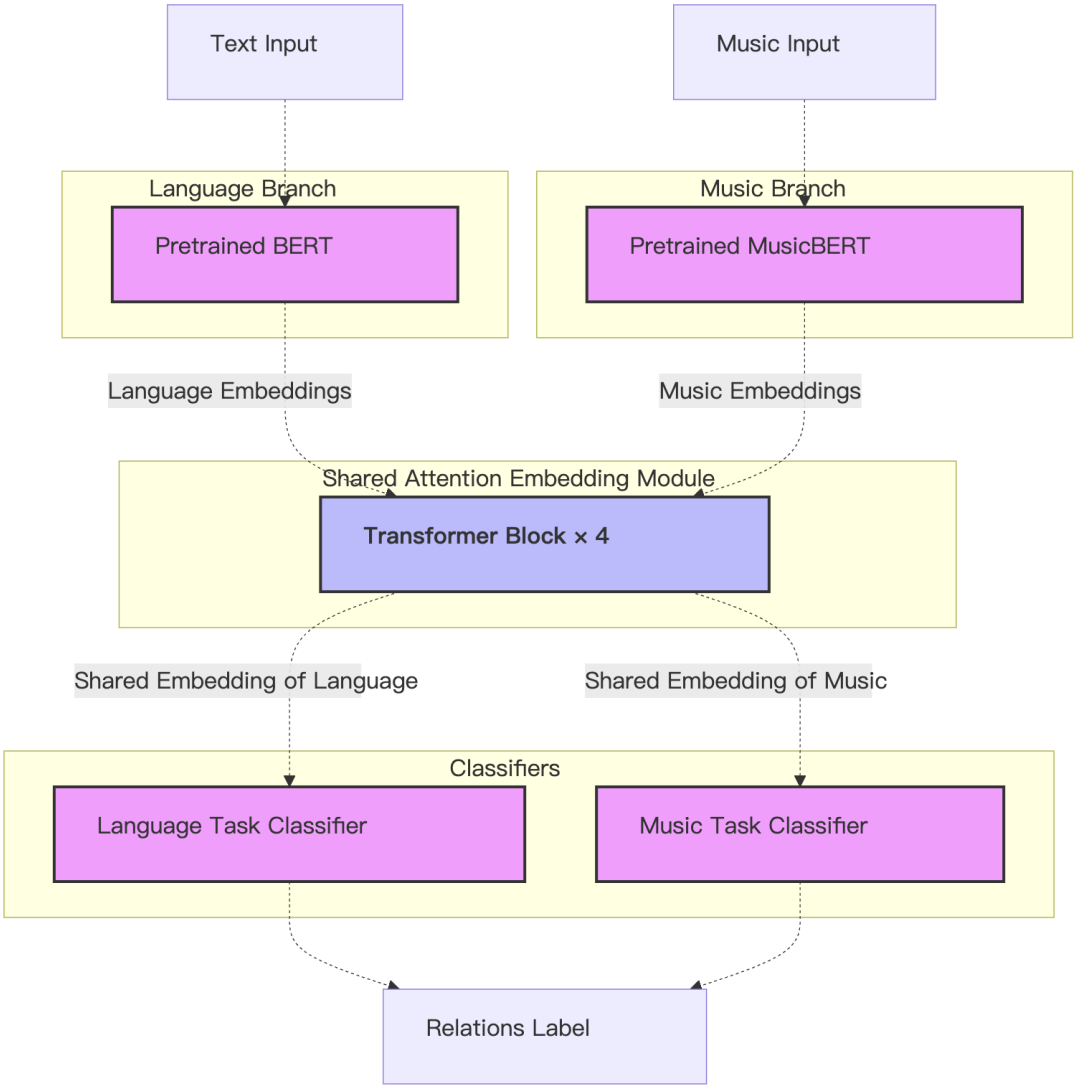


Figure 6: **Model Architecture**.

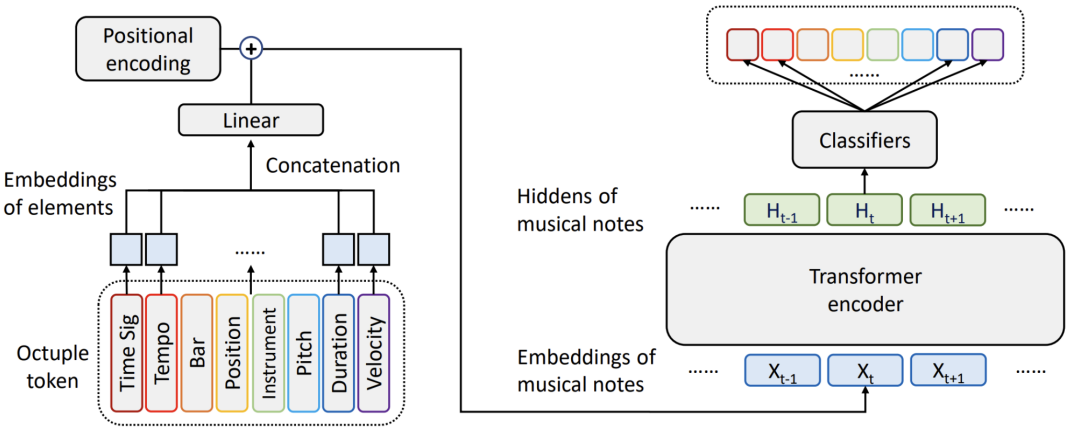


Figure 7: **MusicBERT Architecture[7]**.

Task	Data Size	# Problem Types	Metrics	Answer Type	Text Sources
Foundational Temporal Understanding Tasks					
Ordering	29,462	2	Acc.	3-Way MC	MCTACO[1], Wikipedia, Misc.
Frequency	4,658	6	Acc.	3-Way MC	MCTACO[1], SQuAD[2], Misc.
Duration	7,232	7	Acc.	3-Way MC	Same
Typical Time	13,018	4	Acc.	3-Way MC	Same
Temporal Interpretation and Computation Tasks					
Amb. Res.	3,649	5	Acc.	3-Way MC	Misc.
Arithmetic	15,629	9	Acc.	4-Way MC	Same
Advanced Temporal and Conceptual Understanding Tasks					
Relation	102,462	1	Acc./F1	3-Way MC	TempEval-3[3]
Temporal NLI	282,144	1	Acc./F1	3-Way MC	MNLI[4], SNLI[5]
Causality	1,200	2	Acc.	2-Way MC	COPA[6], Misc.
Storytelling	67,214	1	Acc.	2-Way MC	ROC[7], SCT[8]

Figure 8: **TRAM Overview**. Among all tasks, we only adopt data from the Relation category.