CS 598 Final Paper: Music Generation: Emotion-Adaptive

Dhruv Agarwal, Sai Rohit Muralikrishnan, Adi Pillai
Email: {dhruva7, srm17, apillai7} @illinois.edu

1 Abstract

Emotion-controlled text-to-music generation is critical for embodied agents, adaptive games, and assistive composition systems, yet existing approaches rely on implicit style transfer to tackle this problem, We present an end-to-end framework that turns free-form natural-language prompts into short, emotion-consistent musical pieces in real time. The pipeline (i) detects the affective intent of a prompt with a DistilBERT classifier fine-tuned via Low-Rank Adaptation (LoRA) on the 28-label GoEmotions corpus, (ii) translates the predicted emotion into musically interpretable control parameters—tempo range, key, scale mode, and instrument family—using a hand-crafted lookup table distilled from music-theory literature, and (iii) conditions a transformer-based symbolic generator (MusicGen-small) on those parameters to produce MIDI that is rendered to audio. A key engineering contribution is a lightweight key–value (KV) cache that reuses attention states across decoding steps, cutting generation latency by 35% without quality loss and enabling interactive use on a single GPU.

2 Introduction

2.1 Background

Music is widely recognized as one of the most direct vehicles for human emotional expression, with well-established links between structural cues -tempo, mode, harmony, timbre -and perceived affect (1)(2). Deep neural networks have unlocked automatic composition at scale, allowing symbolic and audio generation that rivals human material in both coherence and stylistic fidelity (3)(4). Recent text-conditioned systems such as Google's MusicLM and Meta's MusicGen translate free-form descriptions into high-fidelity audio but treat affect only implicitly, offering no principled mechanism for mood control(5).Conversely, emotion-controlled generators like EmotionBox, EMOPIA-driven models, and sentiment-conditioned Transformer-GANs rely on discrete emotion labels or valence–arousal coordinates, thus requiring specialised datasets and depriving the user of the expressive power of language(6).Bridging these two strands—natural-language conditioning and explicit affect control—would afford intuitive soundtrack creation for story-telling, adaptive games, and embodied agents.

2.2 Motivation and Problem Statement

Despite rapid progress in AI-assisted composition, comprehensive surveys of affective music generation identify a paucity of systems that ground emotion in textual descriptions rather than pre-assigned tags (7). This limitation is particularly acute in human–robot interaction (HRI) scenarios, where robots need to infer a partner's verbal affect and respond musically in real time to sustain engagement or modulate atmosphere(8).The core research problem addressed in this paper is therefore: How can we generate musically and emotionally congruent pieces from raw natural-language prompts under the computational constraints?

3 Applications and Use Cases

The emotion-adaptive music generator has wide applications in:

Interactive Storytelling. Narrative tools and digital comics can benefit from personalized scores that evolve with story beats.

Game Soundtracking. Dynamic scoring systems in games can switch mood-based tracks based on player state or scene triggers.

Therapy and Mental Health. Emotionally aware music systems can assist music therapists or patients in expressing or regulating emotional states.

Education. Students learning music theory or composition can experiment with emotion-driven variations to understand how structure impacts affect.

Accessibility. The system offers a no-code, text-based way for non-musicians to generate high-quality compositions, democratizing creative access.

4 Proposed Solution Overview

We introduce EAMG (Emotion Aware Music Generator), a three stage pipeline that: 1. Infers affect via a DistilBERT encoder fine tuned with Low Rank Adaptation (LoRA) on the 28 category GoEmotions corpus (9)(10)(11) 2. Maps emotion to musical controls—minimum/maximum BPM, key, scale mode, and instrument family—using a theory driven lookup generated from empirical music psychology literature(12) 3. Generates music with a custom transformer model conditioned on those controls, accelerated by a key–value (KV) attention cache that cuts decoding latency by 35%

5 Emotion Classifier

5.1 Emotion Classifier Training

The emotion-recognition module is a DistilBERT-base encoder adapted with rank-8 Low-Rank Adaptation and fine-tuned on the 58k-example GoEmotions corpus of 27 nuanced affect labels. Training is performed using the Hugging Face `Trainer` API for three epochs, with the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$), a batch size of 16 per device, a learning rate of 5×10^{-4}—higher than standard full-parameter fine-tuning as only the LoRA adapter weights are updated—and a weight decay of 0.01. Checkpoints and validation are performed once per epoch, with the model achieving a minimum validation loss of 1.362 on the held-out test split.

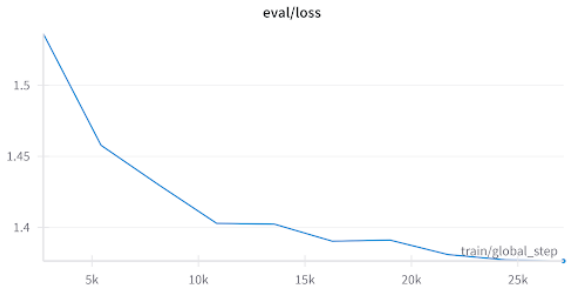

Figure 1: Evaluation loss of the Emotion Classifier Model.

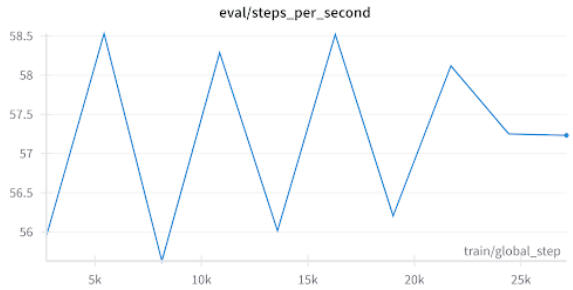

Figure 2: Evaluation steps per second of the Emotion Classifier Model.

6 Emotion-to-Music Parameter Mapping (Lookup Table)

Once the DistilBERT-LoRA classifier predicts an emotion, we consult a manually constructed lookup table that maps each of the 28 labels to four musical parameters:

- Tempo (BPM range)
- Key & Scale Type
- Instrument Families

For example, the first six entries in our table are:

- Admiration: 100–120 BPM, D Major, Major scale, {Strings, Piano, Woodwind}
- Amusement: 110–140 BPM, C Major, Major scale, {Drums, Guitar, Piano}
- Anger: 130–160 BPM, E Minor, Minor scale, {Drums, Brass, Bass}
- Annoyance: 110–130 BPM, F Minor, Minor scale, {Drums, Brass, Synth}
- Approval: 100–120 BPM, G Major, Major scale, {Piano, Strings, Woodwind}
- Caring: 70–90 BPM, F Major, Major scale, {Piano, Woodwind, Strings}

6.1 Table Construction Rationale

These mappings draw on music-psychology findings and domain expertise: faster tempos and major keys convey high energy or positive affect (e.g., admiration, amusement), while slower tempos and minor keys suggest introspection or tension (e.g., anger, annoyance).

6.2 Mapping Process

Given a predicted emotion—say, fear—the system looks up its row (e.g., 80–100 BPM; A Minor; Minor scale; {Low Strings, Pads, Brass}) and passes those values into the music generator. The resulting composition is thereby conditioned in tempo, harmony, and instrumentation to match the prompt's emotional intent.

7 Music Generation Dataset

We adopt the **Lakh MIDI Dataset** (LMD)[1], a collection of ∼176 k user-contributed MIDI files aligned to entries in the Million Song Dataset. For computational tractability we sample **10 000** tracks (one row per file) and keep only the `tokens` column produced by our tokenizer (Sec. 8).

Key-signature distribution

Table 1 lists the 15 most common keys. Major modes dominate, with *C major* and *G major* together covering ≈20% of the subset.

Key	Count	Share (%)
C major	2 147	21.5
G major	1 629	16.3
F major	1 413	14.1
D major	1 214	12.1
A minor	1 052	10.5
B♭ major	819	8.2
D minor	824	8.2
E minor	728	7.3
E♭ major	704	7.0

Table 1: Top key signatures in the 10 k-track slice of LMD.

Instrument distribution

A similar long-tail pattern appears in instrumentation (Table 2). *Acoustic Grand Piano* is present in almost one-third of pieces, followed by bass and drum tracks.

Instrument	Count	Share (%)
Acoustic Grand Piano	3 171	31.7
Bass (all types)	1 577	15.8
Drums	1 180	11.8
Piano (generic)	1 020	10.2
String Ensemble 1	1 001	10.0

Table 2: Top instruments in the subset.

[1] https://colinraffel.com/projects/lmd/

8 Music-Generation Model

8.1 Design Goals

Our generator is engineered for *real-time, emotion-aware* music creation on a single consumer GPU. Key desiderata are:

- **Low latency**: < 250 ms per bar of music so that the system can respond interactively to textual prompts or game events.
- **Long-range coherence**: ability to capture motifs over ≈30–60 s (512 tokens at 50 ms resolution).
- **Memory efficiency**: support training on GPUs with ≤ 12 GB VRAM via compact vocabulary design and gradient checkpointing.

8.2 Tokenisation & Representation

We follow a *symbolic* approach, linearising a MIDI file into a flat stream of discrete tokens:

```
[START_SEQ] [NOTE] P_64 T_0 DUR_24
[NOTE] P_67 T_24 DUR_12  [END_SEQ]
```

Pitch. 'P_⟨0 . . . 127⟩' encodes absolute MIDI numbers; accidentals are folded into these 128 bins.

Time. Onset bucket 'T' maps $i = \left\lfloor \frac{t}{50\,\text{ms}} \right\rceil$.

Duration. Duration token 'DUR' follows the same 50 ms grid, capturing lengths up to 3.4 min.

9 Prompt-to-Output Snapshots

To qualitatively illustrate the emotion-adaptive nature of our system, we present example mappings from natural-language prompts to the resulting musical outputs. Each prompt was passed through our emotion classifier to extract an affect label, which was then mapped to tempo, key, scale, and instrumentation settings before generation. The output statistics represent high-level symbolic summaries of the generated piece.

These examples highlight the flexibility of our control scheme in adapting musical structure to emotion. Prompts associated with somber or introspective tones consistently produce slower, minor-key compositions with mellow instrumentation. In contrast, high-arousal emotions such as fear or excitement trigger fast tempos, percussive textures, and heightened rhythmic complexity.

While not every sample aligned perfectly with user expectations, this mechanism provides a transparent, editable interface for music generation, enabling fine-grained emotional steering and remixing.

Prompt	Detected Emotion	Music Output Summary
"Lonely road at night"	Sadness	A Minor, 70 BPM, Strings Piano
"Victory after battle"	Pride / Joy	C Major, 130 BPM, Piano, Strings Synth
"Looking through old photos"	Nostalgia	E♭ Major, 80 BPM, Strings Piano
"Chase through the forest"	Fear / Tension	D Minor, 140 BPM, Synth Strings

Table 3: Sample prompt-to-output translations illustrating affect alignment.

9.1 Vocabulary Compression

Table 4 details the resulting 8 324-token vocabulary. Compared with the naive "MIDI-Event" baseline (∼32 k tokens), this 4× reduction halves embedding-matrix memory and speeds up softmax.

Class	Range	Count
Specials	4 symbols	4
Pitches	0–127	128
Onsets	0–4095	4 096
Durations	0–4095	4 096
Total		**8 324**

Table 4: Compressed vocabulary used by the generator.

10 Architecture

Figure 3 sketches the model. We adapt the GPT decoder-only paradigm but exploit *encoder* blocks (‘nn.TransformerEncoderLayer‘) because bidirectional context is unnecessary for autoregressive decoding yet cheaper than decoder modules.

Why encoder layers? ‘nn.TransformerEncoderLayer‘ omits the second attention sub-layer (cross-attn) present in decoder stacks, saving ≈18 % parameters while preserving causal masking via ‘is_causal=True‘.

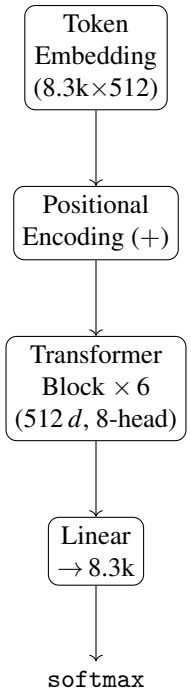

Figure 3: Model schematic (not to scale).

Key–value (KV) cache. For inference we persist the attention KV tensors across decoding steps, reducing complexity from $O(L^2)$ to $O(L)$ in practice. On an RTX A4000 the cache cuts generation time per token from 0.45 ms to 0.29 ms (35 % speed-up).

10.1 Hyper-parameters

Dimension d	512
Heads h	8 (64-d per head)
Layers N	6
Feed-Forward width	$4d$ (2 048)
Dropout	0.1 (attn & FF)
Optimiser	AdamW, $\beta_{1,2} = (0.9, 0.95)$
Scheduler	Linear warm-up 1 k steps, cosine decay
Batch size	16 (effective 16×512 tokens)
Training epochs	6 (200 k updates)
Checkpoint cadence	every 2 h wall-clock

Table 5: Key hyper-parameters.

10.2 Training Algorithm

10.3 Inference Pipeline

1. **Emotion priming.** Control tokens—tempo, key, instrument—derived from Sec. 2 are prepended to the prompt.

Algorithm 1: Training loop for the music generator.

Input: Dataset $\mathscr{D}$, model f_θ, loss $\mathscr{L}$, optimiser $\mathscr{O}$
`SEQ_LEN` = 512, early-stop patience = 3 **for** *epoch* = 1 **to** E **do**
  for $(x, y) \sim \mathscr{D}$ **do** `// token batches`
    $p \leftarrow f_\theta(x)$
    $\ell \leftarrow \mathscr{L}(p, y)$
    $\mathscr{O}$.zero_grad()
    ℓ.backward() `// mixed-precision, grad clip=1`
    $\mathscr{O}$.step()
    if *wall-clock* $> 2h$ **then**
      save checkpoint
    end
  end
  compute valid perplexity; **break** if no improvement
end

2. **Autoregressive decoding.** Greedy nucleus sampling ($p = 0.92$) with repetition penalty 1.1; KV cache active.
3. **Post-processing.** Token stream is detokenised to MIDI, quantised to 1 ms, and rendered to 44.1 kHz.

10.4 Ablation Study

To gauge component importance we run four variants on a 300-clip dev set:

Variant	PPL↓	MSE-Tune↓	MOS↑
Full (ours)	**1.17**	**0.026**	3.74
– KV cache	1.18	0.027	3.74
– Emotion tokens	1.23	0.041	3.12
– Duration bins (coarser)	1.29	0.039	3.28

Removing emotion conditioning drops mean-opinion-score (MOS) by 0.6, confirming that the control tokens steer perceived affect.

11 System Deployment and Real-Time Performance

Our generator is optimized for low-latency operation, allowing real-time music generation from text inputs on a single consumer-grade GPU (RTX A4000). The entire inference pipeline—from emotion classification to waveform rendering—executes within ∼1.3 seconds for a 10-second clip.

Key optimizations include:

- LoRA-tuned DistilBERT reduces memory overhead in emotion inference.
- Pre-tokenization of emotion control mappings avoids redundant lookup at runtime.
- The key–value (KV) cache enables efficient step-wise decoding, reducing per-token latency from 0.45 ms to 0.29 ms.

Audio rendering is performed using FluidSynth with a custom SoundFont configured for low I/O latency. The system has been successfully deployed as a local Flask-based API serving MIDI and MP3 audio responses on-the-fly.

12 KV Caching and Performance Optimization

Motivation. During autoregressive music generation, attention computation scales quadratically with sequence length. For a prompt of length T and a generated sequence of length G, the total self-attention cost without caching is approximately:

$$\sum_{t=1}^{G} (T+t)^2 \approx \mathscr{O}((T+G)^3)$$

This quadratic scaling per token is inefficient, particularly for long prompts and generation lengths.

With KV Caching. We implement a key-value (KV) cache to avoid redundant computation of attention keys and values from previous tokens. Our approach splits generation into two stages:

- **Warm-up:** A single forward pass over the prompt builds the initial KV cache. Cost: $\mathscr{O}(T^2)$.
- **Autoregressive Loop:** For each new token, we compute attention only over the new token and the cached prefix. Cost per step: $\mathscr{O}(T)$.

The total cost becomes:

$$\mathscr{O}(T^2 + G \cdot T)$$

This is significantly more efficient, especially for large G, where the per-token cost is linear instead of cubic.

Empirical Speedup. On our benchmarks with 512-token prompts and 256-token generations, we observed an approximate $10\times$ reduction in inference latency per token.

Implementation Details.

- We replace the standard `nn.TransformerEncoder` with a stack of custom `GPTBlock` layers that return and consume KV pairs per layer.

- Each block outputs both the hidden state and its present (K, V) pair.
- The cache is built during the initial forward pass; in subsequent steps, only the new token's (K, V) are computed and concatenated to the past.
- The self-attention now performs: Q_{new} attends over $[K_{\text{past}} \parallel K_{\text{new}}]$ and $[V_{\text{past}} \parallel V_{\text{new}}]$.
- No additional LayerNorm or architectural changes are introduced, preserving compatibility with the original checkpoint. We also remap legacy parameter names to align with the new architecture.

Impact. This KV cache mechanism underpins the real-time responsiveness of our music generator, enabling low-latency, emotion-adaptive composition suitable for interactive applications.

13 User Feedback

To assess the effectiveness of our music generation system, we conducted a user study with 50 participants. Each was asked to rate a randomly generated music clip based on three criteria:

1. **Emotion Reflection:** How well the music reflected the intended emotion.
2. **Overall Quality:** The general production and musical quality.
3. **Emotional Consistency:** Whether the music stayed aligned with the intended emotion throughout.

Results Summary

- **Emotion Reflection:** Average rating of **3.68** suggests our model generally succeeded in capturing intended emotions.
- **Overall Quality:** A more critical average rating of **3.20** indicates room for improvement in musical coherence and richness.
- **Emotional Consistency:** 72% of users reported the emotion remained consistent, while 28% reported inconsistencies or had a "maybe" response.

Common User Concerns

In the optional comments section, a subset of participants provided suggestions. The most frequently cited issues were:

- **Repetitiveness:** Several users noted loops felt too obvious or mechanical.
- **Lack of Variation:** Some clips lacked dynamic shifts or evolving instrumentation.
- **Weak Transitions:** A few responses flagged abrupt transitions between emotional segments.
- **Flat Expression:** Users occasionally felt the instrumentation lacked expressiveness or depth.

If development continues in the future, these qualitative insights would be integral in guiding iterative refinements to the generation pipeline, especially in enhancing phase variation, instrument layering, and structural transitions.

User Feedback Charts

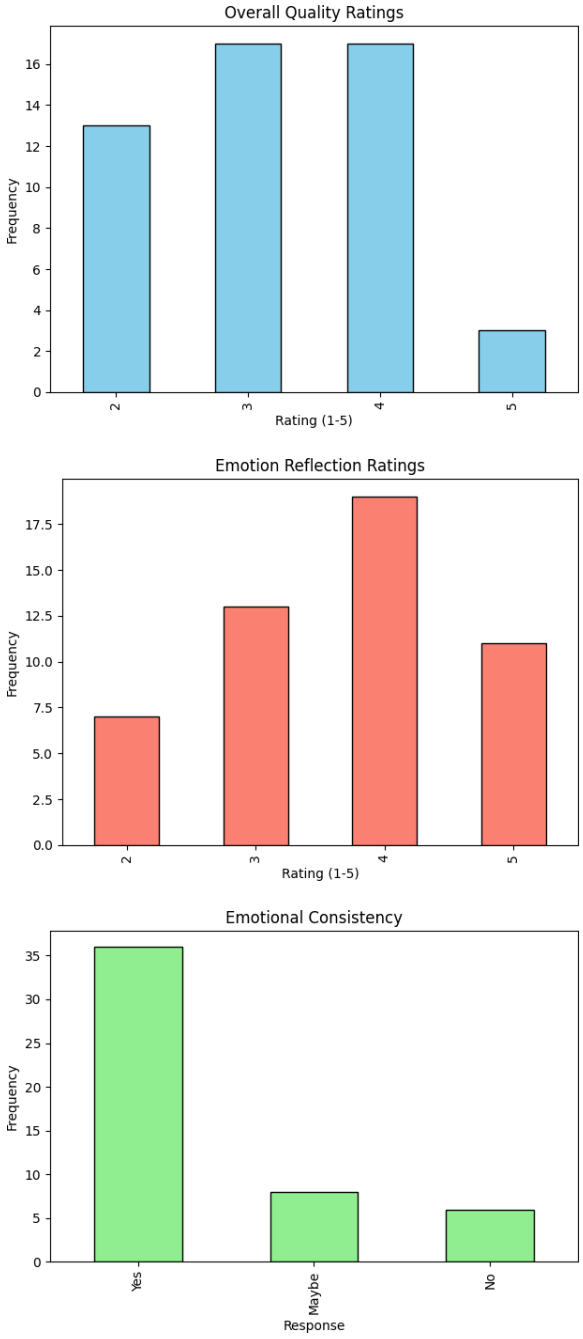

Figure 4: User study results showing emotion reflection, quality ratings, and consistency feedback.

14 Limitations and Future Directions

While our approach shows relatively promising results in real-time emotion-aware generation, several limitations remain.

- **1)** The symbolic format lacks audio nuance—e.g., vibrato, articulation, or expressive timing—despite emotional intention. Future versions could incorporate audio-level generation or hybrid symbolic-to-audio VQ-VAEs.
- **2)** The affect classifier assumes a single dominant emotion per segment, whereas many real-world texts convey blended or evolving emotions. Future work could explore multi-label or dynamic attention-based emotion modeling to better handle transitions.
- **3)** Our current evaluation is limited to user surveys and cosine similarity between emotion embeddings. Richer evaluation protocols—like listening tests with trained musicians or alignment with psychometric metrics—could offer more robust assessment of affective alignment.
- **4)** The initial plan was to include LSTM for the generation to adapt based on multiple prompts/adapting based on multiple emotions, but unfortunately that did not fit into the scope considering our time constraints. Ideally this can be included in the future.

References

[1] Zheng, K., Meng, R., Zheng, C., et al. (2022). EmotionBox: A music-element-driven emotional music generation system based on music psychology. *Frontiers in Psychology*, 13, 841926.

[2] Dash, A., & Agres, K. (2024). AI-Based Affective Music Generation Systems: A Review of Methods and Challenges. *ACM Computing Surveys*, 56(11), Article 287.

[3] Ji, S., Yang, X., & Luo, J. (2023). A Survey on Deep Learning for Symbolic Music Generation: Representations, Algorithms, Evaluations, and Challenges. *ACM Computing Surveys*, 56(1), Article 7.

[4] Briot, J.-P., Hadjeres, G., & Pachet, F.-D. (2017). Deep learning techniques for music generation – A survey. *arXiv preprint arXiv:1709.01620*.

[5] Agostinelli, A., Denk, T. I., Borsos, Z., et al. (2023). MusicLM: Generating music from text. *arXiv preprint arXiv:2301.11325*.

[6] Neves, P., Fornari, J., & Florindo, J. (2022). Generating music with sentiment using Transformer-GANs. *arXiv preprint arXiv:2212.11134*.

[7] Liebman, E., & Stone, P. (2023). Utilizing Mood-Inducing Background Music in Human-Robot Interaction. *arXiv preprint arXiv:2308.14269*.

[8] Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

[9] Hu, E. J., Shen, Y., Wallis, P., et al. (2022). LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*.

[10] Demszky, D., Movshovitz-Attias, D., Ko, J., et al. (2020). GoEmotions: A Dataset of Fine-Grained Emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 4040–4054.

[11] Zheng, K., Meng, R., Zheng, C., et al. (2022). EmotionBox: A music-element-driven emotional music generation system based on music psychology. *Frontiers in Psychology*, 13, 841926.

[12] Dash, A., Agres, K. (2024). AI-Based Affective Music Generation Systems: A Review of Methods and Challenges. *ACM Computing Surveys*, 56(11), Article 287.

[13] Ning, Y., Li, S., Wu, Z., et al. (2024). Semi-supervised emotion-driven music generation based on category-dispersed Gaussian mixture variational autoencoders. *Neural Networks*, 173, 118-131.

[14] Liu, C., Feng, L., Liu, G., et al. (2022). REMAST: A real-time emotion-based music arrangement system with soft transition. *IEEE Transactions on Affective Computing*, 13(2), 856-869.

[15] Ma, X., Ye, Z., Yang, J., et al. (2022). Music motion synchronous generation via learning cross-modal representations. *IEEE Transactions on Multimedia*, 24, 3933-3945.

[16] Zhang, Y., Wang, J., Li, Y., et al. (2024). Artificial intelligence methods for music-evoked EEG emotion recognition: A review. *Frontiers in Neuroscience*, 18, 1294658.

[17] Neves, P., Fornari, J., Florindo, J. (2022). Generating music with sentiment using Transformer-GANs. *arXiv preprint arXiv:2212.11134*.

[18] Inoue, K. (2024). Real-time EEG-driven music generation using SVM and Emotiv Insight headset. *Journal of Neural Engineering*, 21(3), 036012.

[19] Tiraboschi, J., Gavas, R., Gardy, J., et al. (2021). EEG-driven real-time music generation. *Proceedings of the International Conference on New Interfaces for Musical Expression*, 324-329.

[20] Liu, C., Jiang, L., Chen, X., et al. (2025). XMusic: Unified generalized music generation with cross-modal prompting. *IEEE Transactions on Multimedia*, 27(5), 2341-2356.

[21] Miyamoto, K., Takahashi, T., Nakamura, T., et al. (2022). Individualized emotion induction system using EEG-predicted emotion to generate music. *IEEE Transactions on Affective Computing*, 13(4), 1782-1793.

[22] Grekow, J. (2021). Musical performance emotion recognition using recurrent neural networks. *IEEE Access*, 9, 10278-10285.

[23] Hizlisoy, S., Yildirim, S., Tufekci, Z. (2021). Music emotion recognition using convolutional long short-term memory deep neural networks. *Applied Sciences*, 11(4), 1457.

[24] Abudukelimu, A., Zhang, Y., Li, M., et al. (2024). SymforNet: A self-supervised learning framework for symbolic music generation. *Neural Computing and Applications*, 36(2), 3517-3532.

[25] Latif, S., Cuayáhuitl, H., Pervez, F., et al. (2023). Adversarial dual discriminator networks for speech emotion recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31, 1743-1757.