MuseMood:
A Dynamic Multimodal System for Robust Input Fusion

Annie Xu – jx2603
Beijia Zhang – bz2527

— Abstract —

MuseMood is a dynamic multimodal system for music emotion recognition that adaptively handles noisy and unreliable inputs across audio, lyrics, and MIDI. By evaluating input quality and adjusting each modality's weights accordingly, MuseMood achieves stronger performance than traditional static fusion models, particularly in noisy or incomplete scenarios. These findings underscore the value of adaptive fusion strategies for building more reliable and practical emotion recognition systems

COMS E6156 – Topics in Software Engineering
Final Report
Columbia University
May 5, 202

Introduction

In many real-world applications such as emotion recognition and video understanding, AI systems have increasingly relied on multimodal inputs to improve performance and context awareness. These systems combine multiple types of input like text, audio, and visual data to form a more complete picture of the task at hand. By capturing different aspects of the data, multiple modalities work together to improve the overall quality of predictions. However, a common limitation in many existing multimodal models is the assumption that all inputs are equally reliable. Most traditional fusion models apply fixed weights to each modality regardless of how clean or degraded the input actually is [1]. When inputs are affected by real-world issues like background noise, poor video quality, or missing information, the model performances can degrade significantly.

To address this, recent research has started to explore dynamic fusion methods that adjust to the quality of each input. These approaches have led to improvements in accuracy and robustness for tasks like sentiment analysis and emotion classification [2]. Building on this direction, we developed MuseMood, a focused implementation that explores dynamic fusion for music-based emotion recognition. The key component of MuseMood is a Dynamic Modality Weighting Module that checks how reliable each input is and adjusts its impact on the final result. This allows the model to place greater emphasis on cleaner inputs while reducing the influence of those that are noisy or incomplete.

We tested MuseMood under various degraded input scenarios including noisy audio, missing lines in lyrics, and corrupted MIDI sequences. In these experiments, MuseMood consistently outperformed models that used static fusion, demonstrating the effectiveness of its adaptive design. By making the model more responsive to real-world noise and imperfections, we are able to achieve more reliable emotion recognition performance even when some inputs aren't ideal.

Methodology and Metrics

MuseMood is designed to recognize emotions from music using three primary modalities: audio, lyrics (text), and MIDI. It builds upon the MIREX Multimodal Emotion Dataset [3], which contains 764 lyric files (in .txt format) and 196 MIDI files (.mid). Since our system is designed to analyze and dynamically fuse information from audio, lyrics, and MIDI, it requires all three modalities to be present for each sample in order to function properly. Missing any one modality would prevent the model from performing a complete assessment of input quality and hinder the effectiveness of the fusion process. As a result, we restrict our dataset to the 196 songs that include complete data across all three modalities.

To test how well the model performs in more realistic, imperfect conditions, we simulated degradation for each modality. For audio, we added background noise, reduced signal-to-noise ratio (SNR), and lowered energy levels. Lyrics were corrupted by randomly removing lines or replacing words with blanks to mimic incomplete or missing content. For MIDI, we removed notes or altered timing to simulate structural issues. During training and evaluation, only one modality is randomly degraded at a time. This setup makes the model adjust its behavior based on how good the inputs are which helps us see how well our dynamic weighting strategy actually works.

We built two versions of the model and both use the same 3-layer feedforward neural network. We picked this setup due to its simplicity, ease of understanding, and compatibility with our relatively small dataset. The model takes in combined features from audio, lyrics, and MIDI, and then predicts an emotion label.

- ➔ **Baseline Fusion Model**
  The baseline treats all three modalities equally. Their features are concatenated and fed directly into the FNN without considering whether any modality might be noisy or unreliable. This approach assumes clean and complete data across the board which is rarely the case in practice.

- ➔ **Dynamic Fusion Model**
  The dynamic model builds on the baseline by introducing a Dynamic Modality Weighting Module. Before fusion, the system evaluates the quality of each modality in real time and adjusts its weight accordingly. Each feature vector is scaled by a score that reflects how trustworthy that modality is for the given input. Cleaner modalities are given more weight, while degraded ones contribute less to the final prediction. The weighted vectors are then concatenated and passed through the same FNN.

To determine the quality of each modality, we use simple heuristics designed for each data type:

- ➔ For audio, we evaluate SNR and energy. Low values usually suggest noise or silence.
- ➔ For lyrics, we check for missing or blank lines, and measure content length. Sparse or unusually short lyrics are marked as degraded.
- ➔ For MIDI, we look for structural issues like missing notes, uneven timing, or long gaps, which may indicate corruption.

Each modality is assigned a reliability score between 0 and 1. These scores are used to scale the corresponding feature vectors before fusion.

The model is trained using PyTorch with a batch size of 16. We use the Adam optimizer with a learning rate of 0.001 and cross-entropy loss which is appropriate for multi-class emotion classification. The training runs for 50 epochs. During training, we use the degradation metadata to generate reliability scores for each modality and apply them dynamically in every batch. Evaluation is done on the same dataset using standard metrics: accuracy, precision, recall, and F1-score. We evaluate the model using torch.no_grad() to prevent gradient updates and ensure fair assessment. Each test sample includes one randomly degraded modality and we average results across three random seeds for consistency.

Research Questions and Methodology

RQ1: Does integrating dynamically computed modality weights improve classification performance compared to using static weights?

To understand the impact of dynamic weighting, we compared two models with the same architecture but different fusion strategies. One used static fusion, where all modalities are treated equally, while the other used dynamic weighting based on the real-time quality of each modality. As described in the earlier section, we simulated real-world noise by randomly degrading one modality per sample. We evaluated

both models using common classification metrics—accuracy, precision, recall, and F1-score—averaged over three random seeds to reduce variance.

The results clearly show that the dynamic model outperformed the baseline across all metrics:

Metric	Baseline	Dynamic
Accuracy	0.3834	0.4819
Precision	0.4622	0.5178
Recall	0.3834	0.4819
F1-Score	0.3005	0.4769

The dynamic model outperformed the baseline by a wide margin, with the most notable improvement seen in the F1-score, which increased by over 58%. Since the F1-score balances precision and recall, this suggests that the dynamic model was not only more accurate overall but also better at reducing false predictions across all emotion classes. In a task like this, where emotional labels are often subtle and overlapping, such improvements can make a significant difference.

For example, Cluster 1 includes strong, high-energy emotions like boisterous, confident, and passionate, while Cluster 3 focuses on more reflective or bittersweet moods, such as wistful, poignant, or melancholic. Being able to tell these apart, especially when one input modality is degraded, requires the model to make careful use of the remaining data. The dynamic model appears to do exactly that, relying more on cleaner inputs to make better decisions when others are unreliable.

Because both models used the same training process and network structure, this also serves as a kind of ablation study. The performance gap highlights the importance of the dynamic weighting module, not only for improving scores but also for making the model more adaptable and resilient when input quality varies.

RQ2: What is the tradeoff between computational complexity and performance gains introduced by the quality-aware fusion mechanism?

Adding real-time modality weighting does increase computational complexity, since the system has to assess the quality of each input before applying the fusion step. In our case, we used simple, rule-based heuristics to estimate input degradation: for audio, we measured signal-to-noise ratio and energy levels; for lyrics, we checked for missing or incomplete lines; and for MIDI, we looked at structural issues like missing notes or irregular timing. These checks are lightweight and don’t require additional model training, which makes them a practical choice—especially for small datasets or low-resource setups like ours.

Even with this added complexity, the performance improvement is significant. The dynamic model reached an F1-score of 0.4769 compared to 0.3005 for the baseline, marking a 58% gain. This shows that even basic, heuristic-based assessments can make a big difference in how well the model handles noisy or incomplete inputs.

It is also worth noting that our setup was relatively simple: we worked with just 196 fully aligned multimodal samples, and used a 3-layer feedforward neural network. In this context, the overhead from quality-aware fusion was minimal. But in larger or more complex systems, using more advanced quality estimators like trained degradation classifiers or confidence-aware encoders could offer more accurate assessments. That said, these would also increase training time, model size, and inference costs, which weren't feasible for our project's scope. Overall, there's a clear tradeoff: adaptive fusion strategies do boost performance and robustness, but choosing the right method depends on how much computational overhead is acceptable based on the system's goals and available resources.

RQ3: How does the performance of a multimodal fusion model change under modality degradation (e.g., noisy audio, blurred visuals)?

When we tested the baseline model under conditions where one modality was randomly degraded, we saw a noticeable drop in performance. Despite two modalities remaining clean in each sample, the baseline model lacked the ability to identify and discount the unreliable one. As a result, it often made incorrect predictions that achieved only 0.3834 in accuracy and 0.3005 in F1-score.

One common issue we observed was misclassification between emotion clusters. For instance, songs labeled as wistful or poignant (Cluster 3) were frequently mistaken for tense-anxious or intense (Cluster 5), especially when the lyrics were incomplete or the audio was noisy. In many of these cases, the degraded modality was key to expressing the emotion. For example, telling apart emotions like bittersweet and fiery often depends on subtle details in the lyrics. When the text input was incomplete or noisy, the baseline model missed those clues and ended up relying too much on MIDI or audio which does not always capture those subtle emotional differences clearly.

A key reason for the performance drop is that the baseline model does not evaluate the quality of its inputs. It treats all modalities the same, even when some are clearly degraded. In contrast, our dynamic model considers factors like audio noise levels, completeness of the lyrics, and the structure of MIDI notes to determine which inputs are more reliable. These checks helped guide the model to trust cleaner modalities more, leading to better predictions.

This result highlights more than just the limitations of static fusion; it also emphasizes the importance of degradation awareness in multimodal systems. Looking ahead, there is potential to improve even further by exploring more advanced degradation detection techniques, such as confidence scoring from pre-trained models, learned degradation classifiers, or unsupervised anomaly detection. These could make multimodal systems even more robust and better suited to real-world, noisy environments.

RQ4: How does the system perform when one or more modalities are missing entirely at inference time?

MuseMood is built to handle missing or unreliable inputs by dynamically adjusting the weight of each modality during inference. If a modality is completely missing, for example, due to sensor failure or missing data, the system assigns it a low or zero weight which effectively removes its influence from the final prediction. This makes it possible for the model to shift focus to the remaining, more reliable modalities.

While this behavior is supported in our design, our experiments were focused on a more controlled scenario: all three modalities (audio, lyrics, and MIDI) were always present, but one was randomly degraded per sample. This setup allowed us to isolate the model's ability to adapt to noisy inputs without introducing the additional complexity of complete modality absence. Still, the same dynamic weighting mechanism that helps the model deal with degraded inputs could also be applied when a modality is missing altogether. In real-world applications like user-generated uploads with incomplete metadata or field recordings missing audio, we think this kind of flexibility is especially important. Although we didn't explicitly test full modality dropout in this project, MuseMood's architecture is naturally suited to that kind of setup.

A promising direction for future work would be to systematically test the model under complete modality removal. Exploring fallback strategies like modality-specific subnetworks, imputing missing data, or training the model to handle varying input combinations could further improve its robustness in real-world environments where not all data is guaranteed.

Future Work

While MuseMood shows strong potential in handling degraded multimodal inputs through dynamic weighting, there are several areas where the system could be improved in future work. One major limitation is the dataset size, as our experiments were based on only 196 samples that had fully aligned audio, lyrics, and MIDI. This relatively small dataset limits how well the model can generalize to more diverse music styles and emotional expressions. Expanding the dataset to include a wider range of genres, formats, and emotional labels would help the system better handle real-world variability.

Another area for improvement is the type of degradation used. In this project, we relied on rule-based, synthetic degradation to simulate noisy or incomplete inputs. While this allows for controlled testing, it doesn't fully capture the messiness and unpredictability of real-world data. Future work should consider using naturally degraded samples, such as those from live recordings, incomplete uploads, or user-generated content, to better reflect how the system might perform in practice.

There is also room to enhance how the system detects and responds to input quality. Right now, we use simple heuristics to estimate degradation. A more advanced approach would be to train machine learning models to estimate modality confidence directly from the data. This could lead to smarter and more adaptive weighting strategies that go beyond fixed rules.

In addition, MuseMood uses a simple feedforward neural network, mainly so we could focus on testing our fusion method. But in the future, we could try more advanced architectures, like attention mechanisms, transformers, or separate networks for each modality to help the model better understand the complex relationships between the different inputs.

Finally, our system currently classifies songs into broad emotion clusters. A more detailed or multi-label prediction approach could capture a wider range of emotional nuance, such as bittersweet, whimsical, or intense. This would open up new possibilities for applications like emotion-aware music recommendation, mood-based content tagging, and affective computing systems that adapt to how users feel even when inputs are noisy or incomplete.

Overall, these future improvements would help push MuseMood toward a more robust, flexible, and real-world-ready system. Potential applications include emotion recognition from noisy recordings, smarter tagging of music videos, adaptive live performance systems, and more personalized music recommendations that continue to work well even when input quality isn't perfect.

System Deliverables

All system deliverables are hosted at https://github.com/xxanxnie/MuseMood. Please refer to the repo README for detailed setup, usage, and project materials.

Self Evaluation

Annie Xu

This project gave me a much deeper understanding of multimodal learning, especially the challenges that come with handling imperfect data in real-world scenarios. I took the lead on data processing tasks, cleaning and aligning the audio, lyrics, and MIDI inputs, and built the degradation pipeline to simulate noise and corruption across modalities. I also implemented logic to track which modality was degraded for each sample. Through this, I gained practical experience with data preparation, input simulation, and assessing data quality. Beyond that, I contributed to system design discussions and helped analyze the evaluation results. Working on this end-to-end pipeline sharpened my ability to reason through design trade-offs, think critically about performance versus complexity, and develop solutions that work within constraints like latency and incomplete data.

Beijia Zhang

Doing the MuseMood project was a particularly rewarding experience for me and I really learned a lot. Not only did I get a lot of technical improvements, but it also gave me a more practical understanding of how research is done. I was mainly responsible for the model development part, from architecture design to training and evaluation. We made static and dynamic fusion models, and I added a mechanism that can judge the quality of different modalities in real time, and then ran a lot of experiments under different noise conditions to see how stable the models really are. I was also responsible for integrating the various input features, and arranging tests in simulated “bad data” environments to verify the model's resistance. During the whole process, our team worked very closely together, and we have been communicating with each other since the beginning of data cleaning. I think this project made me realize that a truly useful AI system must be able to cope with all kinds of imperfect data situations.

Reference

[1] A. Gandhi, K. Adhvaryu, S. Poria, E. Cambria, and A. Hussain, "Multimodal sentiment analysis: A systematic review of history, datasets, multimodal fusion methods, applications, challenges and future directions," Information Fusion, vol. 91, pp. 424–444, Mar. 2023. [Online]. Available: https://doi.org/10.1016/j.inffus.2022.09.025

[2] M. N. Yeğin and M. F. Amasyalı, "Modality Weighting in Multimodal Variational Autoencoders," in Proc. 2022 Innovations in Intelligent Systems and Applications Conf. (ASYU), Antalya, Turkey, 2022, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9925305

[3] R. Panda, R. Malheiro, B. Rocha, A. Oliveira, and R. P. Paiva, "Multi-modal music emotion recognition: A new dataset, methodology and comparative analysis," in Proc. 10th Int. Symp. on Computer Music Multidisciplinary Research (CMMR), Marseille, France, 2013. [Online]. Available: https://www.kaggle.com/datasets/imsparsh/multimodal-mirex-emotion-dataset