RLHF on conditional music generation

Yun-Han Lan **Fang-Duo Tsai** **Shu-Wen Chen** **Tim Liu**
Nation Taiwan University
National Taiwan University, Taipei 106319, Taiwan
{r10946020,r12942150,r11942176,r11942143}@ntu.edu.tw

Abstract

In today's rapidly evolving field of artificial intelligence (AI), there exist models capable of receiving prompts and generating music as output. However, a persistent challenge arises when these models are tasked with generating high-quality music based on prompts that include specific musical conditions. To address this issue, we propose the use of Reinforcement Learning from Human Feedback (RLHF). We employed DDPO to fine-tune AudioLDM 2, utilizing rewards to train the policy network. With two distinct reward models, we obtained two sets of results. When CLAP was employed as the reward model, all similarities showed improvement. On the other hand, when EMOPIA was used as the reward model, improvement was observed only in the case of 10-second music. Additionally, we discovered that longer music durations, despite lower quality, proved to be more beneficial for training. Another notable finding was that overly complicated prompts negatively impacted the training process.

1 Introduction

Numerous AI models, such as text-to-audio and text-to-music converters, have emerged, capable of generating audio and music based on provided text. An example is MusicGen, a model developed by Meta that utilizes twenty-thousand hours of licensed music as training data and is built upon a Transformer-based Language Model. Despite the existence of various text-to-music models, several challenges persist.

One notable issue arises when the content of the prompt includes specific music conditions; in such cases, text-to-music models often struggle to produce music that satisfies all given conditions. Additionally, challenges related to low generation quality and high computational costs remain to be addressed in this domain. While these models represent significant advancements, it is clear that there is ongoing work required to overcome these challenges and further enhance their capabilities.

AudioLDM2 is one of the models that has attempted to address this problem. AudioLDM2 utilizes a Latent Diffusion Model, which serves to reduce computational costs and enhance music quality. It can be trained on a computer with only one CPU (Central Processing Unit) or GPU (Graphics Processing Unit). During the training phase, AudioLDM2 takes two inputs: a prompt and audio. The prompt undergoes processing through GPT-2 to generate the Language of Audio (LOA) associated with the prompt. Simultaneously, the audio is processed through the AudioMAE-encoder to generate the LOA for the audio. These LOAs are then fed into the diffusion model. A probabilistic switcher controls the probability of the latent diffusion model using both the ground truth AudioMAE and the GPT-2 generated AudioMAE feature as conditions.

The second important component is CLAP (Contrastive Language-Audio Pretraining), which can evaluate the similarity between the prompt and the music. We will treat this similarity as a reward, making CLAP our chosen reward model. CLAP takes a text-music pair as input and then jointly

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

trains the audio and text encoders to learn similarity through contrastive learning.

The final component we require is DDPO (Denoising Diffusion Policy Optimization), which is also a Diffusion Model but incorporates reinforcement learning techniques to enhance performance. DDPO is based on the Stable Diffusion Model, and we employ it to train the policy network. This policy network provides us with a gradient to fine-tune AudioLDM2.

Our architecture is designed as follows: first, input the prompt into AudioLDM2 to generate music; second, utilize CLAP to evaluate the similarity between the prompt and the generated music; lastly, employ DDPO to train the policy network using the reward (similarity) and provide feedback to AudioLDM2.

2 Related Work

2.1 Reinforcement learning

Reinforcement Learning from Human Feedback (RLHF). Large Language Models (LLMs) have made significant strides in recent years in generating diverse text based on human prompts. However, measuring "good" text remains a challenge as it involves subjective judgment and context-dependency. Traditional training methods such as next-word prediction (e.g., cross-entropy) have their limitations, and standard metrics like BLEU or ROUGE offer only simple document comparison. This is where Reinforcement Learning from Human Feedback (RLHF)[1] comes into importance. It optimizes models by directly utilizing human feedback, converting human judgment into reward learning. It enables the application of reinforcement learning to complex tasks that are based on human judgment, allowing LLMs to adapt to a wide range of text data and align with complex human values, opening up new possibilities for the development of language models.

Scaling Reinforcement Learning from Human Feedback with AI Feedback (RLAIF). Reinforcement Learning from Human Feedback (RLHF)[1] is an effective technique for aligning language models to human preferences. However, gathering high-quality human preference labels can be a time-consuming and expensive endeavor. RLAIF[2] uses large language models to generate preference labels, reducing the need for human annotators. Tested across various tasks, RLAIF demonstrated the ability to match and even excel the performance of RLHF, showing its potential to achieve human-level performance.

Denoising Diffusion Policy Optimization (DDPO). Applying Reinforcement Learning (RL) to directly train diffusion models for downstream objectives, such as human-perceived image quality or drug effectiveness, involves interpreting denoising diffusion as a multi-step decision-making process. This interpretation enables the use of a class of policy gradient algorithms called denoising diffusion policy optimization (DDPO)[3]. DDPO is used to refine Stable Diffusion on objectives hard to express via prompting, such as image compressibility, and those derived from human feedback, like aesthetic quality. DDPO also shows the ability to enhance the alignment between prompts and images without human annotations, using feedback from a vision-language model.

2.2 Music audio generation

AudioLDM2. The most important feature introduce in AudioLDM2 is LOA(Language of Audio). It replace embedding in AudioLDM become the intermediate feature. LOA can represent the semantic information of an audio clip no matter it is fine-grained acoustic information or coarse-grained semantic information. It also chnage audio-encoder to AudioMAE(Audio Masked Autoencoder) and change text-encoder to GPT-2(Generative Pre-trained Transformer 2). Using GPT-2 allow AudioLDM2 to input flexible conditions, such as the representation of text, audio, image, video, and so on. It use a switcher to choose audio LOA or condition LOA as input for Diffusion Model. The other parts are similar to AudioLDM, it also use VAE(Variational Autoencoder) to decode sample.

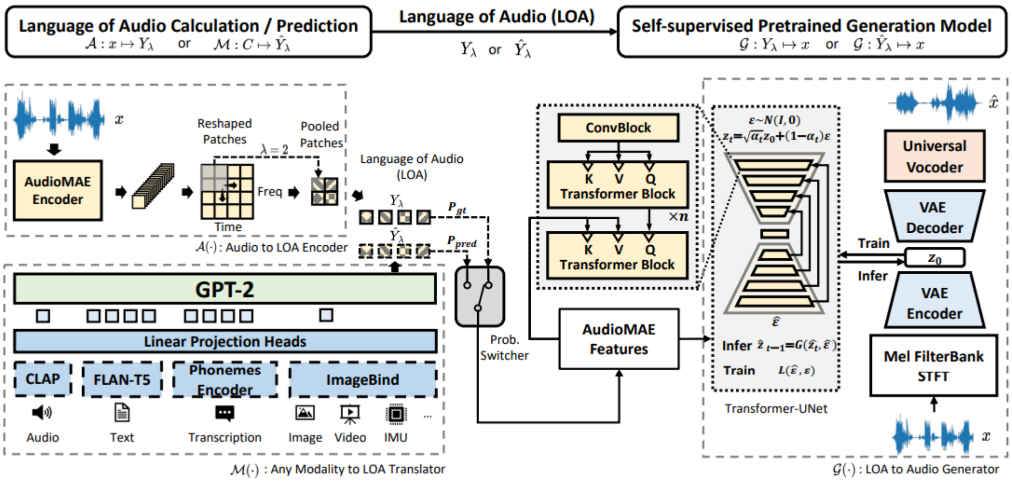


Figure 1: Architecture graph of AudioLDM2

2.3 Audio Feature Extraction

Contrastive Language-Audio Pretrainging (CLAP). CLAP[4] represents a significant advancement in audio classification using contrastive learning. The model is pretrained on three datasets. Initially, LAION-Audio-630K, a substantial dataset, includes 633,526 audio-text pairs gathered from diverse data sources. Secondly, AudioCaps+Clotho (AC+CL) comprises approximately 55,000 training samples of audio-text pairs. Lastly, Audioset consists of 1.9 million audio samples with only labels available for each sample. The dataset comprises a total of about 4 million samples, spanning approximately 30,000 hours, including various genres of music and audio, all accompanied by captions. The proposed pipeline in CLAP incorporates different audio and text encoders, thereby facilitating the development of an audio representation. This design effectively combines audio data with corresponding natural language descriptions, enriching the potential applications in the field.

EMOPIA. EMOPIA dataset[4] is a shared multi-modal (audio and MIDI) database concentrating on the perceived emotion in pop piano music. This dataset was designed to facilitate research on various tasks related to music emotion. The EMOPIA dataset contains 1,087 music clips from 387 songs and clip-level emotion labels annotated by four dedicated annotators. It also provides a short-chunk Resnet model to classify music into four categories.

3 Problem Formulation

In this paper, our objective is to investigate a novel approach for enhancing text-music alignment, specifically focusing on fine-tuning AudioLDM2. AudioLDM2 faces challenges in interpreting conditions within prompts, and our focus will be on addressing the issues outlined below:

- Improve AudioLDM2's capability to understand the meaning of prompts.
- Enhance AudioLDM2's capability to recognize the emotion in prompts.

The following methodology and experiments are designed to address the above two problems with proposed model architecture and different reward models.

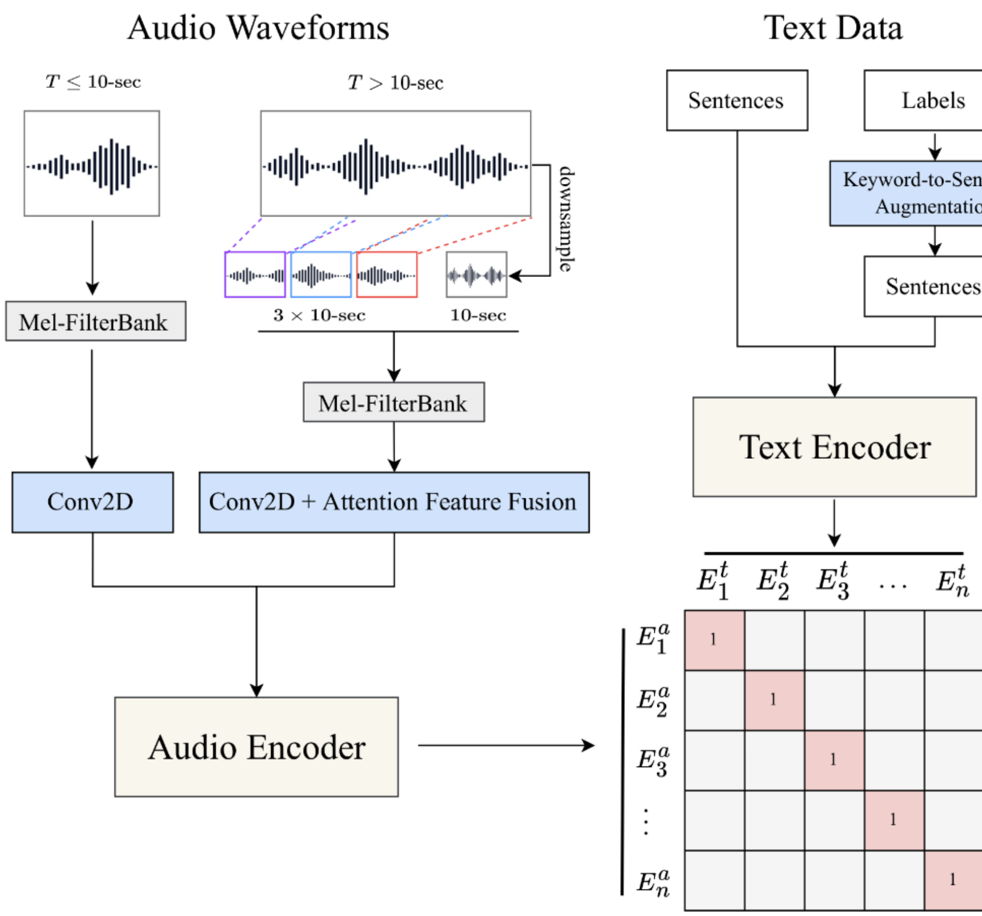


Figure 2: Architecture graph of CLAP

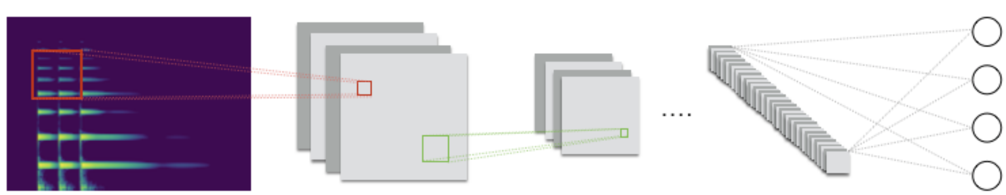


Figure 3: EMOPIA

4 Method

4.1 Model conditioning

We know that the content of prompts significantly influences the outcomes of the diffusion model. To optimize the input prompt format, we experiment with various formats. Our exploration lead us to a successful format as follows:

A recording of an (emotion) (instrument) solo, high quality

In this format, "emotion" can be replaced by one of four emotions: happy, angry, sad, or tender. Similarly, "instrument" can be replaced with a variety of instrument names, such as piano, violin, or flute. Furthermore, we discover that negative prompts notably affect the performance of the diffusion model. Consequently, we test different negative prompt patterns and identify the most effective format:

Low quality, multiple sound sources

The term "low quality" helps steer the diffusion model away from generating low-quality music. As we aim to generate solo instrumentals, the phrase "multiple sound sources" is used to prevent the model from producing music with accompanying instruments or background noise. Notably, we observe a significant performance enhancement in the model after the inclusion of "multiple sound sources".

4.2 Denoising as a Multi-Step MDP

We map the iterative denoising procedure to the following Markov Decision Process (MDP):

$$
\begin{aligned}
s_t &\triangleq (c, t, x_t), \\
\pi(a_t \mid s_t) &\triangleq p_\theta(x_{t-1} \mid x_t, c), \\
P(s_{t+1} \mid s_t, a_t) &\triangleq (\delta_c, \delta_{t-1}, \delta_{x_{t-1}}), \\
a_t &\triangleq x_{t-1}, \\
\rho_0(s_0) &\triangleq (p(c), \delta_T, \mathcal{N}(0, I)), \\
R(s_t, a_t) &\triangleq \begin{cases} r(x_0, c) & \text{if } t = 0, \\ 0 & \text{otherwise,} \end{cases}
\end{aligned}
$$

where x_t is the noisy latent variable, t is the time step, c is the corresponding context, π_t is the policy given the state and action, P is the transition kernel, $\rho_0(s_0)$ is the distribution of initial states, and δ_y is the Dirac delta distribution with nonzero density only at y. Trajectories consist of T time steps, after which P leads to a termination state. The cumulative reward of each trajectory is equal to $r(x_0, c)$. To perform multiple steps with an offline policy, we use an importance sampling estimator. Maximizing the following function is our objective in this MDP:

$$
\nabla_\theta J_{DDRL} = \mathbb{E}\left[\sum_{t=0}^{T} \frac{p_\theta(x_{t-1} \mid x_t, c)}{p_{\theta_{old}}(x_{t-1} \mid x_t, c)} \nabla_\theta \log p_\theta(x_{t-1} \mid x_t, c) r(x_0, c)\right].
$$

The first term represents the difference between the old policy and the updated one; we clip the difference to 1×10^{-4} to avoid drastic changes in the model during offline training.

4.3 Our training pipeline

RLHF[1] can adapt text-to-audio diffusion models to objectives that are challenging to express via prompting, such as audio quality derived from human feedback. However, RLHF requires large-scale human labeling efforts. Motivated by recent work on RLAIF[2], we propose using an existing audio classification model, such as CLAP[4] and EMOPIA[4] to replace additional human annotation.

In Figure 4, we present the architecture of our design aimed at enhancing prompt-audio alignment. This improvement leverages feedback from audio classification models (CLAP & EMOPIA) and utilizes a policy gradient algorithm (DDPO) to update the gradient of the text-to-audio diffusion model (AudioLDM2).

The architecture operates in three distinct steps. Initially, in the first step, we feed the conditional prompt and negative prompt into AudioLDM2 to generate a short segment of music. In the subsequent step, both CLAP and EMOPIA are utilized as individual reward models. When CLAP is employed as the reward model, it extracts embeddings of input text prompt and music, by calculating the cosine similarity of the two embeddings as the reward score. On the other hand, EMOPIA categorizes the music's emotion into four categories, using output logits as the reward score.

In the final step, we gather the output reward scores from the reward models and the log probability between each denoising process, which is considered a multi-step decision-making process, as a

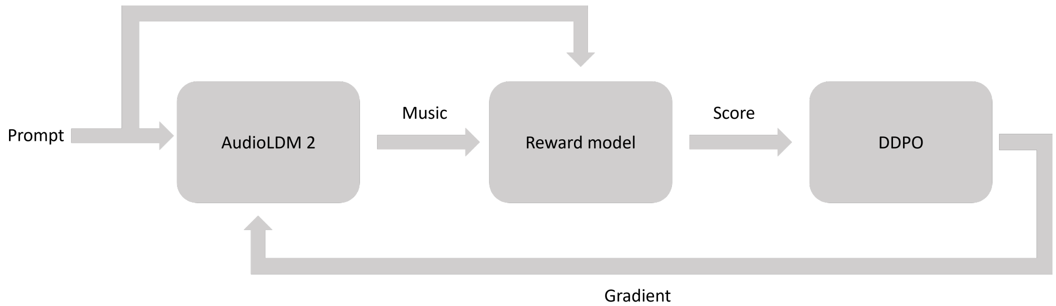


Figure 4: Architecture

trajectory. Subsequently, we train the policy network with each trajectory. The policy network produces gradients used to update the gradient of cross attention layers in AudioLDM2. Additionally, we freeze the majority of the weights in AudioLDM2 while updating the gradient and employ LoRA[5] to fine-tune it. This strategy is designed to reduce the GPU memory requirement, a crucial consideration given we are operating with a single GPU, NVIDIA RTX 3090 resource.

5 Experiment result

In this section, we perform multiple experiments using CLAP and EMOPIA as individual reward models. Our goal is to evaluate the effectiveness of reinforcement learning (RL) algorithms in fine-tuning text-to-audio diffusion models. This fine-tuning aims to enhance the alignment between the input text and the output audio.

5.1 Reward function design

Initially, for both CLAP and EMOPIA reward models, we devise two distinct reward functions: label reward and value reward. The label reward is binary, assigning a value of 0 or 1 based on the correctness of the output prediction. A correct prediction yields a reward of 1, while an incorrect prediction results in a reward of 0. The value reward, on the other hand, ranging from -1 to 0, involves calculating the difference between the probability logit of predicted class and the probability logit of ground truth class as the reward value.

However, after conducting several experiments with these two different reward functions, we found that only the value reward function performed optimally. The experimental results revealed that a dense reward, which is logits score, proved to be much more effective than a sparse reward, which is label-oriented score.

5.2 CLAP model experiments

5.2.1 Random seed design

We discovered that the choice of seed setting significantly affects the performance of DDPO training. Figures 5a and 5b demonstrate that using a random seed results in failed reward training. However, when we used a specific seed (777) under the same experimental conditions, the results (as shown in Figure 5c.) were quite different, with successful training and an average clap similarity of 0.46776447. In an attempt to broaden our testing parameters, we extended the audio duration to 10 seconds and decreased the step sizes to 19. In this scenario, with a random seed setting, the reward curve demonstrated successful training, as displayed in Figure 5d. From our experimental findings, we deduced that a duration of 5 seconds might contribute to less robust training, with only certain seeds yielding success. Conversely, a longer duration seems to enhance training robustness, even with a random seed setting, leading to successful outcomes.

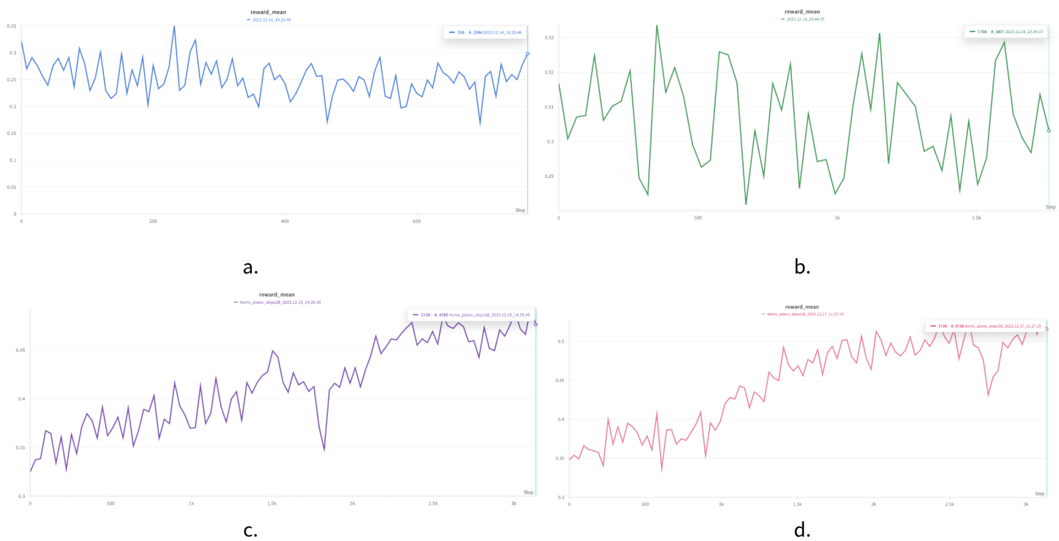


Figure 5: Different seed settings: (a) random seed with 38 sample steps, a batch size of 1, and generates output audio lasting 5 seconds. (b) random seed with 38 sample steps, a batch size of 1, and generates output audio lasting 5 seconds. (c) seed 777 with 38 sample steps, a batch size of 1, and generates output audio lasting 5 seconds. (d) random seed with 19 sample steps, a batch size of 1, and generates output audio lasting 10 seconds.

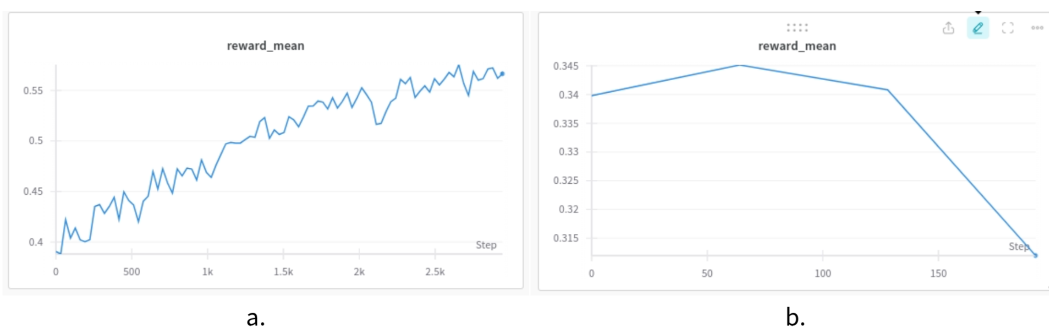


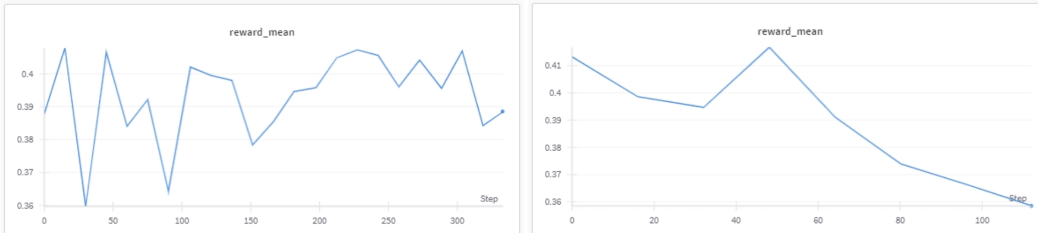
Figure 6: Different instruments prompt settings: (a) single piano prompt with 38 sample steps, a batch size of 1, and a duration of 5 seconds. (b) instruments prompt set with piano and violin, and the configuration is 38 sample steps, a batch size of 1, and a duration of 5 seconds.

5.2.2 Variety of instruments

After multiple experiments, we discovered that using a variety of instruments in prompts resulted in unsuccessful training. Consequently, we scaled down from ten different options to just one, specifically choosing the piano. As depicted in Figure 6a., the use of a single instrument prompt led to successful training, achieving an average clap similarity of 0.4999865. However, when we introduced another instrument to the prompt set, namely the violin, the outcome, as indicated in Figure 6b., resulted in failed training. We speculate that variations in emotional expression between each instrument could potentially confuse the model, ultimately leading to unsuccessful training.

5.2.3 Transfer between reward models

To verify the generalization of the reward models, our initial attempt involved training with CLAP as the reward model. Employing settings of 19 sample steps, a batch size of 1, and a duration of 10 seconds, the outcome yielded an average clap similarity of 0.46776447. However, subsequent



a. b.

Figure 7: Emotional information in diffusion steps experiments: (a) Train only for the final 20 denoising steps. (b) Train only for the initial 20 denoising steps.

evaluation using EMOPIA showed an accuracy of only 0.27. Remarkably, this accuracy closely mirrored the score obtained before applying DDPO for fine-tuning, which was 0.26. Furthermore, we conducted training with EMOPIA as the reward model and assessed it using CLAP for evaluation. The outcome remained consistent with the previous attempt, indicating the inability of this algorithm to generalize to other metrics.

5.2.4 Emotional information in diffusion steps

Intrigued by the question of where emotional information might exist within the denoising steps, we divided the denoising process into two segments. We conducted training for the final 20 steps and the initial 20 steps separately. Both experiments maintained the same settings as those in Figure 6a. experiment, with 38 sample steps, a batch size of 1, and a duration of 5 seconds. These conditions had previously yielded successful training. However, as depicted in Figures 7a. and 7b., the outcomes of both experiments were unsuccessful. From these results, we infer that emotional information cannot be trained solely on specific steps; instead, it necessitates training across all steps.

5.3 EMOPIA model experiments

5.3.1 Training

Initially, we start with the experiment setting as described in Figure 9a, which includes 16 emotions and a single instrument (piano) in the input prompt. The experiment uses 16 sample steps, a batch size of 16, and generates output audio lasting five seconds. However, the results indicate that the training reward curve did not show improvements with this setup.

Given the complexity of handling multiple emotions, we modified the input prompt to include only the four basic emotions: happy, angry, sad, and tender, as shown in Figure 9b. Even with this simplification, the results remained unchanged.

Subsequently, we hypothesized that an increase in step sizes might enhance the quality of the audio output from AudioLDM2. Therefore, we adjusted the experimental setting described in Figure 9c. Due to GPU memory limitations, the step sizes were set at 38, and the batch size was decreased to one. We again used a broader range of emotions, back to 16 in total. However, this adjustment did not lead to an improvement in the results.

Finally, we modified the setup to extend the audio duration to ten seconds, as depicted in Figure 9d. With GPU memory limitations in mind, we reduced the step sizes to 19 with a batch size of one. The results, after 3,000 training steps, revealed an improved reward of -0.1. This setting demonstrated our model's ability to enhance alignment between input text and output audio.

Our analysis across these four experiments concludes that a 5-second duration may lead to failed training. This could be due to the EMOPIA model, which takes 3-second chunks for classification. A 5-second output only provides one chunk, potentially leading to a lack of robustness.

5.3.2 Accuracy on Four Emotions

For a deeper understanding of EMOPIA's performance, we conducted an analysis on its emotion-specific performance. As indicated in Table 1, we observed that prior to fine-tuning AudioLDM2 with

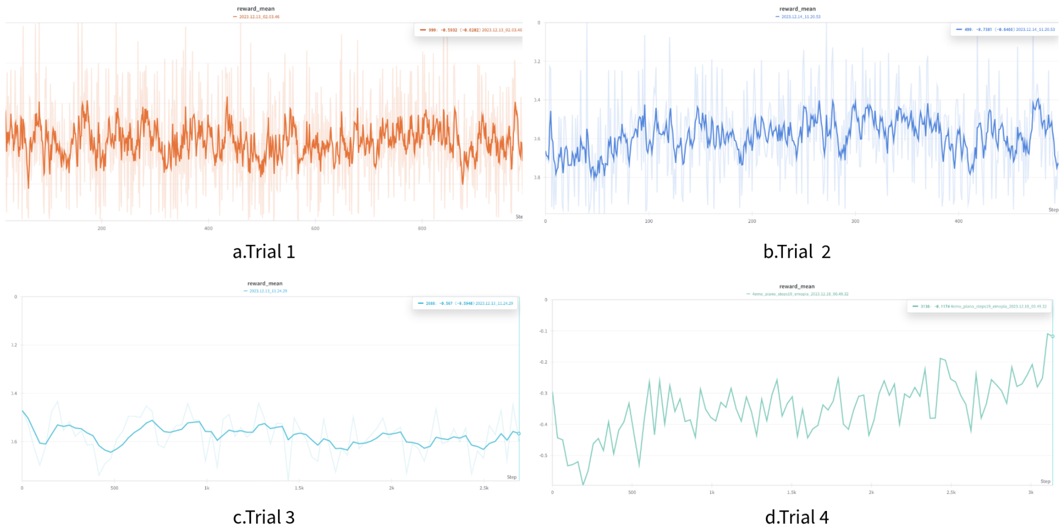


Figure 8: We conducted experiments with various settings involving prompts and the configuration of AudioLDM2: (a) Input prompt includes 16 emotions with a single instrument (piano). The experiment uses 16 sample steps, a batch size of 16, and generates output audio lasting 5 seconds. (b) Input prompt includes 4 emotions with a single instrument (piano). The experiment uses 9 sample steps, a batch size of 16, and generates output audio lasting 5 seconds. (c) Input prompt includes 16 emotions with a single instrument (piano). The experiment uses 38 sample steps, a batch size of 1, and generates output audio lasting 5 seconds. (d) Input prompt includes 4 emotions with a single instrument (piano). The experiment uses 19 sample steps, a batch size of 1, and generates output audio lasting 10 seconds.

Table 1: Accuracy on four emotions

	Original			Trained
Setting	19-step, 10 sec.	38-step, 5 sec.	200-step, 5 sec.	19-step, 10 sec.
Happy	1.0	0.92	0.76	0.6
Angry	0.04	0.0	0.08	0.6
Sad	0.0	0.16	0.08	0.24
Tender	0.0	0.08	0.0	0.0

DDPO, the EMOPIA model tended to classify all audio generated from different emotion prompts as "Happy", irrespective of the settings used. This resulted in the "Happy" emotion prompt exhibiting the highest accuracy, while the other three emotions displayed disastrously low accuracy. However, upon the application of DDPO for the fine-tuning of AudioLDM2, we noticed significant improvements in the classification of the "Angry" emotion, with the accuracy increasing from nearly 0.0 to 0.6. We also recorded a slight improvement for the "Sad" emotion, with its accuracy increasing to 0.24. Despite this, the accuracy of the "Happy" emotion decreased to 0.6, while the "Tender" emotion's accuracy remained stagnant at 0.0. The results provide compelling evidence that our method, which involves using DDPO to fine-tune AudioLDM2, is effective in enhancing the alignment between different emotion prompts and their corresponding audio.

5.4 Results

In summary, our method has demonstrated notable improvements in enhancing prompt-audio alignment. As illustrated in Table 2, when we utilized CLAP as the reward model in a setting with 19 sample steps, a batch size of 1, and a duration of 10 seconds, we observed a slight increase in similarity, from 0.43 to 0.46, before and after training. A more noticeable improvement occurred in

Table 2: Experiment result summary

	CLAP(similarity)		EMOPIA(accuracy)	
Model	Original	Trained	Original	Trained
19-step, 10 sec.	0.43	0.46	0.26	0.36
38-step, 5 sec.	0.33	0.45	0.29	X
200-step, 5 sec.	0.31	-	0.23	-

the setting with 38 sample steps, a batch size of 1, and a duration of 5 seconds, showing an increase from 0.33 to 0.45, nearly matching the performance of the longer duration setting.

The same level of improvement was noticed when we used EMOPIA as the reward model. In a configuration of 19 sample steps, a batch size of 1, and a duration of 10 seconds, the accuracy rose by 10 percent, from 0.26 to 0.36. However, due to EMOPIA's lesser robustness, we were unable to train the model in a 5-second duration setting. Additionally, given the GPU memory limitations, we were unable to train the model using 200 sample steps. Nevertheless, we observed that increasing the number of sample steps did not enhance the performance in both the CLAP and EMOPIA results.

6 Conclusions

We demonstrate the feasibility of employing reinforcement learning (RL) to train text-music alignment. We present a method using DDPO to fine-tune AudioLDM2, and our experiments suggest that this pipeline can improve AudioLDM's ability to recognize emotions and meaning in prompts. However, certain defects require further investigation. As observed, five seconds of music can only be trained on a specific seed, while ten seconds of music can be trained on a random seed. We suspect this discrepancy may be due to EMOPIA, which usually categorizes music as happy and occasionally provides rewards that are meaningless for training. Another potential reason is that EMOPIA uses three seconds as a chunk, so five seconds of music only has one chunk, whereas ten seconds of music has three chunks, making it easier for training.

Additionally, overly complicated prompts may result in training failures. In our experiments, we restricted the use to four emotions and piano, resulting in satisfactory outcomes. However, introducing more complex emotions and instruments led to failures. Another observation is that the model does not transfer seamlessly. Specifically, if we train AudioLDM2 with CLAP and then evaluate it with EMOPIA, the results are the same as if it had not undergone training, and vice versa.

7 References

References

[1] Daniel M Ziegler et al. "Fine-tuning language models from human preferences". In: *arXiv preprint arXiv:1909.08593* (2019).

[2] Harrison Lee et al. *RLAIF: Scaling Reinforcement Learning from Human Feedback with AI Feedback*. 2023. arXiv: 2309.00267 [cs.CL].

[3] Kevin Black et al. *Training Diffusion Models with Reinforcement Learning*. 2023. arXiv: 2305.13301 [cs.LG].

[4] Yusong Wu et al. *Large-scale Contrastive Language-Audio Pretraining with Feature Fusion and Keyword-to-Caption Augmentation*. 2023. arXiv: 2211.06687 [cs.SD].

[5] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL].

[6] Hsiao-Tzu Hung et al. *EMOPIA: A Multi-Modal Pop Piano Dataset For Emotion Recognition and Emotion-based Music Generation*. 2021. arXiv: 2108.01374 [cs.SD].

[7] Benjamin Elizalde et al. *CLAP: Learning Audio Concepts From Natural Language Supervision*. 2022. arXiv: 2206.04769 [cs.SD].

[8] Yuntao Bai et al. *Constitutional AI: Harmlessness from AI Feedback*. 2022. arXiv: 2212.08073 [cs.CL].