Supplementary Information for: The Amateur-Professional Semantic Divide: A Hidden Barrier to AI Music Generation's Alignment and Diversity

1 Personnel Organization, Task Division, Quality Assurance Mechanisms, and Review Requirements

- **Professional Group.** Annotates structures, rhymes, and provides professional descriptions.
- **Amateur Group.** Provides colloquial descriptions.
- **Inspection Group.** Evaluates structural annotations and rates music descriptions.
- **Administrator.** Handles and provides feedback on inquiries from all groups and conducts random checks on the deliverables of each group.

The **grouping** and **training methods** for each group are detailed below.

1.1 Personnel Grouping and Training

1.1.1 Grouping

In the structural annotation phase, which consists of Type A tasks, each data point requires two annotations. In contrast, the music description annotation phase, which consists of Type B tasks, requires only one annotation per data point. Consequently, there are fewer participants in the latter phase. The musical section annotation task in the lyric annotation phase requires basic knowledge of music theory. Therefore, only 104 professionals participated in this task. Among them, 11 highly proficient and responsible individuals were selected as quality assurance inspectors through resume screening and further evaluation. The remaining 93 served as annotators.

In the music description annotation phase, 109 amateur enthusiasts formed the amateur group, and the 93 professionals from the previous phase formed the professional group. Additionally, the 11 inspectors from the previous phase continued as inspectors in this stage. Besides the roles of annotators and quality assurance inspectors, we also selected a member from our research team who was proficient in platform usage, highly professional, and had strong communication skills to serve as the platform administrator.

1.1.2 Training

Next, we provide training for annotators and quality assurance inspectors tailored to their respective roles. First, each annotator logs into CaiMAP to pre-annotate a small dataset of about 20 entries, covering both Type A and Type B tasks. This stage aims to familiarize annotators with the platform's features and to learn the correct procedures for completing annotation tasks. Additionally, we provide specialized training to address common errors, such as removing irrelevant information from lyric texts and accurately identifying each interjection.

On the other hand, the training process for inspectors is more complex. They not only need to be proficient in using the platform but also must establish a unified set of evaluation criteria. We collect the data annotated by the annotators during the pre-annotation phase and distribute the same dataset to all inspectors. For the lyric annotation phase, inspectors must select the annotation they believe to be correct based on the guidelines, or provide an alternative correct annotation if they find the existing ones inaccurate. In the music description annotation phase, inspectors independently evaluate each annotation. After the inspectors complete their tasks, we compile the scores for all music descriptions and organize an inspector meeting. In this meeting, we identify instances where scores from different inspectors vary significantly (with a maximum difference exceeding 10 points) and encourage the inspectors to discuss and agree on a unified evaluation standard. This training process is repeated until the inspectors' scores for the same dataset show substantial consistency.

The next phase is annotation. We have designed an innovative multi-person, multi-stage assurance method aimed at enhancing annotation quality and maximizing its accuracy. Furthermore, this method helps in objectively evaluating the performance of the annotators.

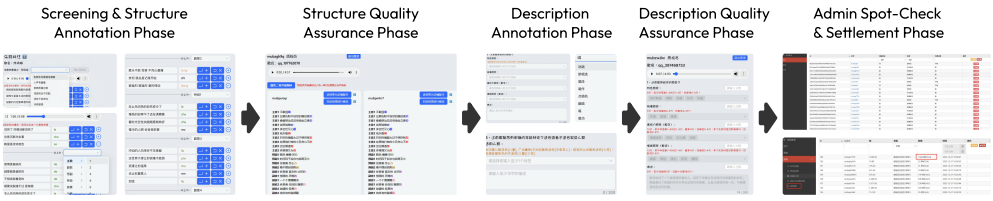


Fig. 1: Data Annotation and Assurance Pipeline. Each annotated data point goes through 5 complex phases to ensure accuracy. The figure shows actual screenshots of each phase's page.

1.2 Quality Assurance Mechanism

In this section, we detail the quality assurance mechanism, including task classification, scoring guidelines, and personnel classification.

Classification	Accuracy(%)
I	[90, 100]
II	[70, 90)
III	[60, 70)
IV	[0, 60)

Table 1: Classification of Annotators of Type A Tasks Based on Accuracy

Classification	Score
I	[90, 100]
II	[70, 90)
III	[60, 70)
IV	[0, 60)

Table 2: Classification of Annotators of Type B Tasks Based on Score

Dimension	Score	Standard
Expressive Impact (S. & A.)	13	4 for Number of Labels; 4 for Label Relevance; 5 for Innovation
Emotional Impact	13	4 for Number of Labels; 4 for Label Relevance; 5 for Innovation
Textual Description	8	3 for Description Relevance; 5 for Word Counts and Innovation
Musical Genres	8	8 for Level of Detail
Tempo and Rhythm	5	5 for Label Relevance
Instrumentation	12	5 for Number of Labels; 3 for Label Relevance; 2 for Description Relevance; 2 for Description Thoroughness
Song Purpose	6	3 for Label Relevance; 3 for Innovation
Culture and Region	6	3 for Label Relevance; 3 for Innovation
Target Audience	6	3 for Label Relevance; 3 for Innovation
Vocal Components	12	5 for Number of Labels; 3 for Label Relevance; 2 for Description Relevance; 2 for Description Thoroughness
Audio Effects	5	5 for Label Relevance
Lyric Themes	6	3 for Label Relevance; 3 for Innovation
Total	100	-

Table 3: Scoring Guidelines of Professional Music Description

Dimension	Score	Standard
Perception of Uniqueness	8	4 for Label Relevance; 4 for Innovation
Perception of Tempo	5	3 for Label Relevance; 2 for Innovation
Expressive Impact (S.)	13	4 for Number of Labels; 4 for Label Relevance; 5 for Innovation
Emotional Impact (L.)	13	4 for Number of Labels; 4 for Label Relevance; 5 for Innovation
Textual Description	8	3 for Description Relevance; 5 for Word Counts and Innovation
Instrumentation	12	5 for Number of Labels; 3 for Label Relevance; 2 for Description Relevance; 2 for Description Thoroughness
Song Purpose	6	3 for Label Relevance; 3 for Innovation
Culture and Region	6	3 for Label Relevance; 3 for Innovation
Target Audience	6	3 for Label Relevance; 3 for Innovation
Vocal Components	12	5 for Number of Labels; 3 for Label Relevance; 2 for Description Relevance; 2 for Description Thoroughness
Audio Effects	5	5 for Label Relevance
Lyric Themes	6	3 for Label Relevance; 3 for Innovation
Total	100	-

Table 4: Scoring Guidelines of Amateur Music Description

1.2.1 Annotation Task Classification

We classify annotation tasks into two categories based on whether they can be objectively evaluated: **Type A**, which can be objectively assessed, and **Type B**, which are subject to subjective evaluation. This section provides examples of the classification for each annotation task. To maximize the accuracy and comprehensiveness of each song's annotation, we assign two annotators to Type A tasks and one annotator to Type B tasks. These tasks are performed separately, not concurrently. In addition, besides the annotators, several quality assurance inspectors are required to evaluate the annotators' output. Based on the Type A and B division, we consolidate Type A tasks into a phase called the **Structural Annotation Phase** and Type B tasks into

a subsequent phase called the **Music Description Annotation Phase**. Data must pass through these two phases sequentially to be included in the dataset. That is, data must first undergo structural annotation and pass quality assurance before proceeding to the music description annotation phase, after which data that passes the music description quality assurance can be added to the dataset. For Type A tasks, if two annotators provide the same annotation, we consider the annotation accurate. However, when discrepancies arise, a quality assurance inspector must render their judgment to determine which result is correct or provide their own accurate annotation if both are incorrect. For Type B tasks, quality assurance inspectors are required to give a score from 0 to 100 for the annotation results, with scoring guidelines detailed in Table 3 and Table 4.

1.2.2 Personnel Classification

To ensure that annotators perform their duties diligently, we have implemented a screening mechanism. In the structural annotation phase, the accuracy of Type A task annotations is evaluated through the aforementioned quality assurance system. In the music description annotation phase, given that Type B tasks involve subjective descriptions that are difficult to objectively evaluate, we randomly select 20% of each annotator's annotations for quality control review. In addition, we also assess certain behaviors indicated by backend analytics, such as the frequency of interaction with the progress bar and task skipping. Annotators who demonstrate perfunctory engagement will be warned. In both phases, annotators are categorized into four classes based on their weekly accuracy rate or average score, as detailed in Table 1 and Table 2. Category IV annotators, as well as those who receive two or more warnings, will be excluded from future tasks, and their data for that week will be disregarded. Category I annotators will receive rewards, while Category III annotators may be penalized.

1.2.3 Other Quality Assurance Measures

Annotators are responsible for screening the data (Type A and B). For songs containing non-Chinese languages, poor sound quality, or content unsuitable for inclusion in the dataset such as pornography or violence, annotators can flag for exclusion and skip their annotation.

When annotating Type A musical sections, annotators must repeatedly listen to a piece of music. Therefore, their level of engagement with the annotation task is assessed by the time they spend on the annotation page, the frequency of their interactions with the progress bar, and the frequency of their clicks on the play/pause buttons.

In textual description annotation (Type B), to ensure that annotators listen to each song carefully and provide thoughtful music descriptions, we stipulate that annotators must first listen to the entire song once before they can adjust the progress bar and playback speed. They must write a textual description of no less than 50 words and are prohibited from writing the description within the first 30 seconds of the song's playback, as well as from copying and pasting any content.

2 Preliminary Experiments and Evaluation of Embedding Models for the RAG Retrieval Library

This section provides supplementary details for the pre-experiments on embedding model selection, corresponding to the main text's section on "RAG (Retrieval-Augmented Generation) System Construction and Application."

2.1 Evaluation Metrics for Cross-modal and Text-to-Text Pre-trained Models

2.1.1 Metrics for Evaluating Text-to-Audio Cross-modal Pre-trained Models

Text-to-audio contrastive pre-trained models play a crucial role in enabling generative models to understand descriptions. They ensure that input text descriptions are converted into dense vector representations that capture semantic nuances.

We employ a retrieval-based metric: given a description, we calculate the top-K retrieval accuracy from a pool of N candidate audio clips. The specific calculation method can be referenced from [1].

2.1.2 Metrics for Evaluating Text-to-Text Pre-trained Models

Considering that models like StableAudio use text-to-text models such as T5 [2] to understand prompt descriptions, we also designed a text-retrieval-based metric. Given an amateur description, we calculate the top-K retrieval accuracy from a pool of N candidate professional descriptions.

The process begins by constructing a pool of N candidate professional descriptions, which is obtained by applying the text-to-text pre-trained model to the professional description data in the test set. Subsequently, we calculate and rank the cosine similarity between the provided amateur description and each of the N candidate professional descriptions.

If the ground truth professional description (GT) belonging to the same song as the input amateur description ranks within the top-K, we consider the retrieval successful. Given M different amateur descriptions, the retrieval accuracy RA is defined as:

$$f_n = \begin{cases} 1, & GT \text{ in the top-}K \text{ place} \\ 0, & GT \text{ not in the top-}K \text{ place} \end{cases} \tag{1}$$

$$RA = \frac{1}{M} \cdot \sum_{i=1}^{M} f_n(i) \tag{2}$$

where $f_n(i)$ indicates if the i^{th} amateur description is successfully retrieved.

2.2 Evaluation of Text-to-Audio Cross-modal Embedding Models

To construct the text-to-audio retrieval library for the RAG system, we conducted a series of pre-experiments aimed at evaluating and selecting the optimal text-to-audio cross-modal embedding model. The core objective of these experiments was to assess the model's ability to accurately retrieve the corresponding song (or its key acoustic features) from the MuChin dataset after receiving colloquial descriptions from amateur users.

Before formally comparing various text-to-audio cross-modal models, we first conducted a series of preliminary fine-tuning experiments on the CLAP (Contrastive Language-Audio Pretraining) model itself to optimize its performance on our specific task and data, and to determine the best training configuration. To enhance the cross-modal alignment capability between text descriptions and audio content (which is crucial for cross-modal encoders), we explored the impact of different training data lengths (short, medium, and long descriptions) and different training strategies (MiniBatch vs. standard Batch size) on the fine-tuning effectiveness of the CLAP model. At this stage, only amateur descriptions from the MuChin dataset were used as input for evaluation.

Training Data		**Laion-Clap**	**Batch**			**MiniBatch**		
Test Data			**S-M-L**	**M-L**	**Short**	**S-M-L**	**M-L**	**Long**
Short	Top-5↑	16.07%	14.56%	13.81%	17.87%	21.32%	21.77%	**24.02%**
	Top-10↑	26.13%	26.58%	22.07%	30.78%	37.69%	37.09%	**38.29%**
	Top-15↑	35.14%	33.78%	22.83%	39.94%	50.00%	48.20%	**50.75%**
Medium	Top-5↑	20.12%	18.92%	15.02%	17.12%	24.17%	26.58%	**27.93%**
	Top-10↑	32.88%	31.08%	27.03%	29.58%	39.79%	**41.59%**	40.39%
	Top-15↑	41.44%	43.99%	37.24%	39.64%	53.00%	50.90%	**53.15%**
Long	Top-5↑	17.12%	13.36%	13.81%	13.66%	18.92%	20.42%	**20.72%**
	Top-10↑	24.92%	24.62%	22.67%	22.52%	28.53%	**33.78%**	32.59%
	Top-15↑	31.53%	31.53%	30.48%	28.83%	38.14%	41.29%	**41.30%**
S-M-L	Top-5↑	17.77%	15.61%	14.21%	16.22%	21.47%	22.92%	**24.22%**
	Top-10↑	27.98%	27.43%	23.92%	27.63%	35.34%	**37.49%**	37.09%
	Top-15↑	36.04%	36.43%	30.18%	36.14%	47.05%	46.80%	**48.4%**

Table 5: Preliminary evaluation results (Top-K accuracy) of the CLAP model fine-tuned on the MuChin dataset with different training strategies (MiniBatch vs. standard Batch) and different amateur description lengths (short, medium, long, mixed). S refers to word-level annotations, such as labels and tags. M refers to short phrase descriptions. L refers to long sentence descriptions. For each test setting, the best result is highlighted in bold.

As shown in the preliminary experimental results in Table 5, the CLAP model fine-tuned on the MuChin dataset using the MiniBatch strategy (CLAP-MiniBatch-MuChin) consistently outperformed the version using the standard Batch strategy (CLAP-Batch-MuChin) across all evaluation metrics. This performance improvement

may be attributed to the more frequent parameter updates in MiniBatch training, allowing the model to more finely adapt to and learn the dynamic characteristics of the descriptive text. Furthermore, we observed that when trained on datasets containing longer descriptions, the fine-tuned CLAP model exhibited stronger performance, outperforming those trained on datasets containing only short, medium, or mixed-length (short-medium-long) descriptions. This suggests that longer descriptions typically contain richer semantic information and contextual cues, which helps the model better learn the complex correspondences between text and audio.

Based on the optimal settings determined from these preliminary explorations—namely, using long amateur user descriptions from the MuChin dataset combined with a MiniBatch training strategy—we conducted more extensive formal experiments. We compared several mainstream text-to-audio cross-modal models, including CLAP (using the optimized configuration), MuLan, UnIVAL, and our in-house CLMP model. To evaluate the performance gains from targeted fine-tuning on the MuChin dataset, we tested both the original (base) and the fine-tuned versions of these models.

Models	**UnIVAL**		**MuLan**	**CLAP**		**CLMP**	
	Base	**MuChin**	**MuChin**	**Base**	**MuChin**	**Base**	**MuChin**
Top-5↑	17.27%	23.78%	25.44%	20.12%	**27.93%**	20.43%	26.65%
Top-10↑	26.19%	35.01%	38.37%	32.88%	**40.39%**	33.52%	39.26%
Top-15↑	33.83%	44.36%	49.25%	41.44%	**53.15%**	42.10%	50.59%

Table 6: Top-K accuracy comparison of different text-to-audio cross-modal models (base versions vs. versions fine-tuned on the MuChin dataset with optimized configuration) for song retrieval based on long amateur user descriptions on the unseen test set of the MuChin dataset. The base version of the MuLan model was not evaluated due to the lack of an official open-source checkpoint.

The evaluation results in Table 6 clearly indicate that the performance of all models fine-tuned on the MuChin dataset surpassed their respective original baseline versions. This demonstrates that leveraging the MuChin dataset for fine-tuning effectively enhances the ability of text-to-audio cross-modal models to understand amateur user descriptions and accurately retrieve matching song audio. Among them, the CLAP model, fine-tuned with the optimized configuration, excelled across all metrics, particularly achieving a Top-15 accuracy of approximately 53.15%, and was therefore selected as the core embedding model for our text-to-audio retrieval library.

2.3 Evaluation of Text-to-Text Embedding Models

To build the text-to-text retrieval library for the RAG system (retrieving professional descriptions from amateur descriptions), we evaluated the performance of three widely used text-to-text pre-trained models: Multilingual-T5 (mT5)[3], RoBERTa[4],

and BERT-base. The evaluation also included the baseline performance of these models with their original weights and the performance improvements achieved after fine-tuning them on the MuChin dataset.

The evaluation results presented in Table 7 show that the performance of all models significantly surpassed their respective original baseline versions after being fine-tuned on the MuChin dataset. This fully demonstrates that fine-tuning on the MuChin dataset (which contains parallel corpora of amateur-to-professional descriptions) can significantly improve the ability of these text models to understand amateur users' colloquial descriptions and accurately retrieve the most semantically similar professional descriptions. Among all tested models, the fine-tuned T5 model performed the best, achieving a Top-15 accuracy of 89.43%, and was therefore selected as the embedding model for our text-to-text retrieval library.

Models	**T5**		**RoBERTa**		**BERT**	
	Base	**MuChin**	**Base**	**MuChin**	**Base**	**MuChin**
Top-5↑	60.21%	**82.55%**	56.44%	75.52%	51.49%	69.98%
Top-10↑	65.50%	**86.22%**	60.11%	80.21%	55.67%	73.75%
Top-15↑	68.33%	**89.43%**	63.67%	82.74%	59.03%	77.96%

Table 7: Top-K accuracy comparison of different text-to-text models (base versions vs. versions fine-tuned on the MuChin dataset) for retrieving corresponding professional descriptions using long amateur user descriptions as input on the unseen test set of the MuChin dataset.

3 Detailed Parameter Settings for Targeted Data Training

In the three-stage targeted data training process and subsequent inference applications described in "4.2 Implementation and Application of Targeted Data Training Method," we adopted corresponding parameter settings for different open-source models and training tasks. The general principle was to follow the original open-source implementation of the models while optimizing and adjusting based on our specific goals (particularly improving the understanding and alignment with amateur user descriptions).

3.1 General Hardware Infrastructure

The main model training and fine-tuning tasks were completed on a high-performance computing cluster equipped with 64 NVIDIA A100 GPUs. Each A100 GPU has 80GB of VRAM. Within the computing nodes of the cluster, 8 GPUs are interconnected at high speed via NVLink technology, while different nodes communicate through an InfiniBand network to ensure the efficiency and scalability of large-scale distributed training.

3.2 Parameter Considerations for Training Stage 1 (Foundational Module Construction) and Training Stage 2 (Lyrics-to-Song Generation)

These two optional stages are primarily for building or enhancing the model's foundational capabilities.

3.2.1 Specific Fine-tuning Parameters for the Lyrics Large Language Model (Lyrics LLM)

- **Base Model:** We selected the Qwen-14B-Chat-Int4 model[5] as the foundation, which has excellent Chinese understanding and generation capabilities. Int4 indicates its 4-bit quantized version, aimed at improving inference efficiency.
- **Efficient Fine-tuning Technique:** QLoRA[6] technology was used for parameter-efficient fine-tuning.
- **Distributed Training and Optimization:** The second stage of DeepSpeed ZeRO optimization technology (ZeRO-2) was used in combination to further optimize memory and VRAM usage.
- **Training Time and Checkpoints:** The saving period for each checkpoint was approximately 22 hours, and the total effective training time (cumulative pure computation time) was about 11 days.
- **Quantization and Alignment:** For models using pre-quantized architectures like Int4, no additional quantization is needed after fine-tuning. We ensured that the fine-tuning data and process were consistent with the model's expected chat or instruction templates.

3.2.2 General Parameters for Other Foundational Modules and Lyrics-to-Song Models

- **Model Architecture and Parameter Count:** For the open-source audio generation models used in this study, we configured them based on their publicly released or best-performing larger-scale versions:
  - **YuE Model:** We used its 7B (7 billion) parameter version based on the LLaMA2 architecture, which employs a two-stage language model design (Stage-1 for music language modeling, Stage-2 for residual modeling).
  - **MusicGen:** We used its 3.3B (3.3 billion) parameter version, which is a single Transformer language model.
  - **Stable Audio Open:** We used its released open-source version, with a total system parameter scale of about 1.32B (1.322 billion), where the core Diffusion Transformer (DiT) component has about 1.057B parameters, the autoencoder has about 0.156B parameters, and the T5 text encoder has about 0.109B parameters.
  - **AudioLDM 2:** We referenced its "Large" version (e.g., AudioLDM 2-AC-Large or AudioLDM 2-Full-Large), with a total parameter scale of about 0.712B (712 million), based on the Latent Diffusion Model (LDM) framework, which combines AudioMAE for extracting "Language of Audio" (LOA) features, GPT-2, and a Transformer-UNet diffusion model.

- **Input and Audio Representation:** Each model adopted corresponding strategies for audio representation based on its selected configuration:
  - **YuE Model (7B version)** uses X-Codec as its primary audio tokenizer, a codec that fuses semantic information (based on HuBERT) with acoustic details. Its codebook-0 is particularly important for capturing key semantic information such as melody and vocal content.
  - **MusicGen (3.3B version)** uses EnCodec to encode 32kHz mono audio into a discrete audio token stream at a 50Hz frame rate, using 4 codebooks (each with a size of 2048).
  - **Stable Audio Open (total system parameters 1.32B)** utilizes its VAE-style autoencoder to compress 44.1kHz stereo audio into a 64-dimensional continuous latent representation, with an effective latent rate of about 21.5Hz.
  - **AudioLDM 2 (Large version, 0.712B parameters)** converts audio (resampled to 16kHz in experiments) into "Language of Audio" (LOA) feature sequences through its AudioMAE component and uses a VAE to process Mel-spectrograms to obtain latent acoustic representations for the diffusion model. The specific dimensions, sampling rate processing, etc., of the audio tokens or latent spaces in these models are determined by the pre-trained audio codec components or their original papers/release specifications.
- **Position Encoding:** To better handle music sequences and text descriptions of varying lengths, we extensively used Rotational Position Embedding (RoPE) technology.
- **Optimizer and Learning Rate:** The standard AdamW optimizer was a common choice, with the initial learning rate tuned in the range of 1×10^{-5} to 5×10^{-4} based on model and dataset size, complemented by learning rate warm-up and linear or cosine decay strategies.
- **Batch Size:** The batch size per GPU was set between 4 and 32, depending on the model size and GPU memory constraints.
- **Training Duration:** The pre-training duration for these stages varied from several days to several weeks, depending on the model and data scale.

3.3 Parameter Considerations for Training Stage 3 (Core Targeted Fine-tuning)

This stage uses the MuChin dataset (excluding the test set) for fine-tuning and is key to achieving alignment with amateur users.

- **Learning Rate Adjustment:** The learning rate was set smaller than in the pre-training phase, for example, in the range of 5×10^{-6} to 1×10^{-4}.
- **Conditional Input:** The "user type" (amateur/professional) label was used as an explicit condition, incorporated into the model input by concatenating it with the Description Embedding to guide the model to learn the distinction.
- **Data Balancing:** Although the amateur and professional descriptions in the MuChin dataset are paired, we paid attention to balancing the opportunities for both types of user data when constructing training batches and adjusted the sampling strategy based on validation set performance.

3.4 Data Preprocessing and Augmentation

In all training stages, we performed rigorous preprocessing on the input data.

- **Audio Processing:** Included resampling to a uniform sampling rate (e.g., 44.1kHz or 48kHz), conversion to mono, and amplitude normalization.
- **Text Processing:** Included text cleaning (removing irrelevant characters), tokenization (for Chinese), and converting descriptive texts and lyrics into the ID sequences required by the model.
- **Data Augmentation:** To enhance the model's generalization ability, we used data augmentation techniques during training, such as random cropping of audio clips and synonym replacement or minor rewriting of text descriptions.

These parameter settings provided a solid technical foundation for our targeted data training method, enabling it to effectively utilize the characteristics of the MuChin dataset to improve the alignment of AI music generation models with amateur users.

4 Detailed Explanation of Targeted Data Training and Inference Strategies

This appendix elaborates on the specific strategies for the three-stage targeted data training process and the subsequent inference application phase, as described in "4.2 Implementation and Application of Targeted Data Training Method."

4.1 Training Stage 1: Optional Foundational Module Construction

This stage is dedicated to building or optimizing the model's foundational components to support subsequent song generation tasks.

4.1.1 Refinement of the Lyrics Large Language Model (Lyrics LLM) and Structured Lyrics Generation

To enhance the model's ability to generate structured lyrics (which serve as important input for subsequent song generation) based on user descriptions, we optimized the Qwen-14B-Chat-Int4 model[5]. Specifically, we used QLoRA[6] as the Parameter-Efficient Fine-Tuning (PEFT) method. The training data included not only thematic information extracted from lyrics but also manually annotated musical sections (e.g., `<verse>`, `<chorus>`, `<bridge>` tags) and rhyming structures. After fine-tuning, this Lyrics LLM is capable of generating lyrics with clear musical structure markers based on input professional or amateur descriptions. These structure markers are designed to match the musical sections of the songs generated by the subsequent music generation model.

4.1.2 Training Strategy for Cross-modal Encoders

To enable various cross-modal encoders to effectively process text descriptions and audio clips of different lengths and achieve effective text-audio contrastive pre-training, we used description texts of varying lengths and corresponding audio segments from

the MuChin dataset during their training. This strategy enhances the encoders' adaptability to the variable inputs of the real world.

4.2 Training Stage 2: Optional Lyrics-to-Song Generation Capability Building

This stage aims to enable the model to master the core capability of generating songs with vocals from lyric text, especially for models that do not natively support this function.

We used approximately 1.5 million deduplicated proprietary songs for supervised pre-training of the model. First, we extracted lyric text from the audio of these songs using Automatic Speech Recognition (ASR) tools, thereby creating a large-scale "lyric text-song audio" paired dataset.

In terms of training strategy, we considered that while lyric timestamps can be used to align audio windows with lyrics during training, in practical inference, lyrics generated by users or LLMs often lack precise timestamps. This makes it difficult for the model to accurately match lyric length with the audio window. The traditional Text-to-Speech (TTS) strategy of predicting duration based on word count faces greater challenges in singing due to its vast variability in rhythm and speed. Therefore, we chose not to use a window-based audio generation method, but rather to use complete audio segments (without random segmentation) during training to generate the entire song at once. To accommodate different song lengths, we applied padding at the end of each audio segment for length normalization. This method offers two key advantages:

- Compared to window-based generation, it can better capture the overall musical structure and coherence of the song.
- During the training process, it no longer requires timestamped lyric data (like .LRC format); regular plain text lyric data (like .TXT format) is sufficient, which reduces the complexity of data preprocessing.

4.3 Training Stage 3: Core Targeted Fine-tuning: Generating Songs from Fused Descriptions and Lyrics

This stage is the most critical for achieving alignment with amateur user expectations, and all open-source models participating in targeted training must undergo this fine-tuning stage.

We used the MuChin dataset (excluding the test set) as the training data for this stage. This dataset contains paired amateur and professional user music descriptions, structured lyrics, and corresponding audio segments. The goal of fine-tuning is to enable the model to generate songs that meet the expectations of users (both professional and amateur) based on the provided descriptions and lyrics.

To ensure consistency between the training process and actual inference scenarios and to enhance the model's generalization ability, we randomly segmented the professional and amateur description texts in the training set into various input lengths, ranging from single keywords to complete descriptive paragraphs. This approach enables the model to more effectively handle user text inputs of varying lengths in practical applications. The core idea is that we used "user type" (amateur/professional) as

an explicit conditional label, incorporating it into the Description Embedding, which is then fed into the model along with lyric information and noise samples. This guides the model to learn and differentiate the semantic expressions of different user groups and adjust its generation strategy accordingly.

4.4 Inference and Application Stage Strategy

After completing the above three training stages, the model enters the inference, application, and evaluation stage.

During inference, the system receives a colloquial description from an end-user (professional or amateur). First, the Lyrics LLM, refined in Training Stage 1, generates structured lyrics based on this description. Subsequently, various pre-trained or fine-tuned cross-modal encoders convert the user's original descriptive text into a semantic embedding vector. This description embedding, along with the embedding of the generated structured lyrics (and necessary noise samples), is input into the music generation model that has undergone the core fine-tuning of Training Stage 3. The model outputs a latent space representation of the audio, which is finally converted by the corresponding decoder into a complete song audio waveform. We use this process to generate samples and conduct comprehensive experimental evaluations using the independent test set of the MuChin dataset.

5 Evaluation Results of Generative Model Variants Fine-tuned on Other Description-Audio Datasets

To further validate the unique advantage of the MuChin dataset in enhancing the model's alignment with amateur user descriptions, we conducted a comparative experiment. We used DiT (Diffusion Transformer) as the default base model architecture. In the pre-training phase, all model variants used the same dataset of 1.5 million proprietary songs. The key difference was in the fine-tuning phase: different model variants were fine-tuned on different text prompt-audio datasets, including the MusicBench [7] dataset, the AudioSparx[1] dataset, and the MusicCaps [8] dataset. We focused on evaluating the objective performance of these models when processing amateur user descriptions.

Datasets	**Automatic Anno-based**		**Manual Anno-based**	
	AudioSparx	**MusicBench**	**MusicCaps**	**MuChin**
SAA↑	0.30	0.34	0.32	**0.44**
ARA↑	0.29	0.36	0.31	**0.57**

Table 8: Evaluation results of generation model variants fine-tuned on different description prompt-audio datasets.

[1] https://www.audiosparx.com

The experimental results in Table 8 clearly show that the model fine-tuned using the MuChin dataset significantly outperformed models fine-tuned on other datasets (such as AudioSparx, MusicBench, MusicCaps) across all alignment metrics for amateur descriptions, more effectively capturing and realizing the intent of amateur users. One possible explanation for this advantage is that the test data used for our objective evaluation metrics (the independent test subset of the MuChin dataset) shares the same domain characteristics as the MuChin fine-tuning data (i.e., both contain real amateur and professional user descriptions). However, it must be emphasized that this study is the first to systematically focus on and differentiate between professional and amateur music descriptions. Therefore, there are currently no other recognized benchmark test sets specifically for this "descriptive divide" for cross-validation. The performance differences among the datasets may also stem from their annotation characteristics: for example, the AudioSparx and MusicBench datasets primarily rely on automated algorithms for tagging to generate music descriptions. There remains a significant semantic gap between these machine-generated labels and the complex, diverse colloquial descriptions of real humans (especially amateur users). Meanwhile, although the MusicCaps dataset contains manual annotations, its annotations are primarily from professional musicians, which leads to its relatively poor performance in aligning with the non-professional descriptions of amateur users. These comparisons further highlight the value of the MuChin dataset in bridging the amateur-professional semantic divide.

6 Detailed Evaluation Metrics for the Lyrics LLM (Used in Preliminary Experiments to Determine the Fine-tuned Qwen as the Optimal Model for Supporting Lyric Generation)

We utilize MuChin to evaluate the capabilities of existing LLMs in generating structured lyrics, including Qwen [9], Baichuan-2 [10], GLM-130B [11], and GPT-4 [12]. Furthermore, considering that Qwen is primarily trained on a Chinese corpus and performs well in a Chinese language context, we further fine-tuned Qwen with another batch of data. Subsequently, we evaluated the performance of this fine-tuned Qwen model on MuChin to assess the efficacy of the data in fine-tuning language and music models, as well as the fine-tuned model's proficiency in understanding music descriptions and performing related tasks.

6.0.1 Results

Table 9 presents the similarity scores in various dimensions for structured lyrics generated by the selected LLMs in a one-shot scenario, using music descriptions as the given prompt. It is noteworthy that all models achieved commendable results. We can observe that among the base models, the overall score increases with the expansion of the parameter scale. Benefiting from its vast parameter scale and extensive training data, GPT-4 significantly outperforms the other three models in most dimensions. However, the fine-tuned Qwen, despite having fewer parameters, markedly surpasses

Model		GPT-4	GLM-4	Baichuan-2	Qwen	
					Base Model	Fine-tuned
Parameter Size		**1800B**	**130B**	**53B**	**14B**	**14B**
Overall Score		67.08(±6.23)	54.93(±16.46)	49.19(±15.85)	48.31(±13.39)	**85.24(±11.65)**
Structure Similarity	Song Level	2.50(±1.16)	2.29(±0.97)	2.32(±0.99)	2.58(±1.51)	**4.69(±2.38)**
	Section Level	32.40(±0.41)	28.20(±6.75)	28.83(±8.02)	26.49(±4.92)	**32.14(±0.91)**
	Phrase Level	15.52(±2.19)	12.93(±4.31)	12.74(±4.36)	11.59(±3.80)	**17.01(±0.80)**
	Word Level	0.36(±0.79)	0.15(±0.39)	0.01(±0.02)	0.10(±0.23)	**9.12(±5.92)**
Rhyming	Fitting Accuracy	13.88(±3.05)	9.61(±5.17)	4.84(±4.72)	8.01(±4.36)	**16.30(±2.94)**
	Proportion Reasonableness	2.40(±2.66)	1.74(±2.65)	0.45(±1.96)	1.29(±1.89)	**5.98(±4.03)**

Table 9: Evaluation results of selected LLMs on the structured lyric generation benchmark. Larger values indicate higher similarity to the corresponding dimensions of the actual lyrics, suggesting better quality of the generated structured lyrics. For the base models, the highest score in each dimension is underlined.

the non-fine-tuned base model in overall score and shows a significant lead in every dimension. This highlights the significant impact of fine-tuning in enhancing the model's ability to understand music descriptions and generate structured lyrics. It also suggests that current LLMs have considerable potential for improvement in the music domain, emphasizing the importance of MuChin in advancing the development of Chinese LLMs in this field.

6.1 Evaluation Metrics for Structured Lyric Generation

In assessing the performance of LLMs, we prompt them with music description inputs, asking for structured lyrics that include musical sections and rhymes. While lyrical content should present subjective diversity, structural integrity remains an objective quality. Hence, our evaluation primarily centers on the accuracy of the lyric structure rather than its content. We introduce an evaluation method that measures the likeness between the model-generated lyrics and the ground truth across the six dimensions outlined below.

- **Song Level:** Measures the similarity between the generated lyrics and the ground truth in terms of overall structure.
- **Section Level:** Measures the similarity between the generated lyrics and the ground truth in terms of musical section labels, order, and the number of sections.
- **Phrase Level:** Measures the similarity in the number of phrases within each musical section compared to the ground truth.
- **Word Level:** Measures the similarity in the word count of each corresponding phrase between the generated lyrics and the ground truth.
- **Rhyming Fitting Accuracy:** Measures the degree to which the generated lyrics match the ground truth in terms of end-of-line rhymes.
- **Rhyming Proportion Reasonableness:** Evaluates the reasonableness of the rhyming proportion in the generated lyrics based on the proportion of rhyming lines within the overall lyrics.

To quantify these structural similarities, we employ the Gestalt pattern matching algorithm [13]. The final score is derived from these dimensions, where the first five are

combined using a cumulative weighting scheme, and the sixth serves as an additional reward score.

6.1.1 Scoring Formulas

The similarity of the overall structure and the musical section structure is calculated according to Equation 3, where K_m represents the number of matching characters in the longest common subsequence between strings A and B. L_A denotes the length of string A, and L_B denotes the length of string B. In the context of the overall structure, A and B represent the entire set of lyrics. In the context of musical section structure, A and B refer to the sequence of musical section labels.

$$p = \frac{2K_m}{L_A + L_B} \tag{3}$$

The within-section structure similarity is calculated according to Equation 4. In this equation, each element of $ListA$ and $ListB$ represents the number of sentences contained in each matching musical section of song A and B, respectively. For example, '[4, 8, 4]' indicates that the three matching musical sections contain 4, 8, and 4 sentences, respectively.

$$p = \frac{2\sum \min(ListA, ListB)}{\sum ListA + \sum ListB} \tag{4}$$

Similarly, the within-sentence structure similarity (word count per phrase) can also be calculated using Equation 4, where each element of $ListA$ and $ListB$ represents the number of words in each matching sentence. The rhyming similarity follows Equation 3, where K_m is the number of matching rhyming lines, and L_A and L_B are the total number of lines.

Since each finer structure depends on the matching of the preceding one, a cumulative similarity is used to account for the influence of macroscopic structures on microscopic ones. Let the similarities of the first five dimensions (song, section, phrase, word, and rhyme fitting) be p_1 to p_5, with respective weights w_1 to w_5. The overall similarity is calculated using Equation 5.

$$p = \sum_{i=1}^{5} w_i \prod_{j=1}^{i} p_j \tag{5}$$

The base score is obtained by multiplying the overall similarity p by 100. This is then combined with a reward score, as detailed below.

6.1.2 Reward Score for Rhyming Reasonableness

An additional reward score is calculated to assess the reasonableness of the rhyming proportion. This mechanism, detailed in Algorithm 1, assigns points based on how the proportion of rhyming lines in the generated lyrics compares to typical structures. In this algorithm, `max_equ_slc_sum` denotes the maximum number of matched phrases; `rc_ing` denotes the number of matched rhyming phrases; `acmp_sr` is the cumulative

Algorithm 1 Reward Score Algorithm

Input: max_equ_slc_sum, rc_ing, acmp_sr, rc_ino
Parameter: EXTRA_POINTS
Output: extscore

```
if max_equ_slc_sum == 0 then
    r_ratio = 0
else
    r_ratio = rc_ing / max_equ_slc_sum
end if
extscore = EXTRA_POINTS * acmp_sr
if 0.6 <= r_ratio and r_ratio <= 0.8 then
    extscore *= 1.0
else if rc_ino == rc_ing and rc_ino > 0 then
    extscore *= 0.7
else
    r_delta = |r_ratio − 0.7|
    if r_delta <= 0.3 then
        extscore *= 0.4 * (1 - r_delta)
    else
        extscore *= 0.0
    end if
end if
return extscore
```

product of similarities from the first five dimensions ($\prod_{j=1}^{5} p_j$); `rc_ino` is the rhyming ratio within the given rhyme scheme; and `EXTRA_POINTS` represents the total possible reward score.

7 Evaluation of Music Understanding Models (Used in Preliminary Experiments to Select MERT-95M as the Audio Embedding Model for Supporting Music Diversity Metrics APED and CD, and Intent Fidelity Metric ARA)

7.1 Music Understanding Models

Similar to pre-trained language models in natural language processing, such as BERT [14], a proficient pre-trained music understanding model should be able to effectively represent information across various dimensions of music, allowing the use of a simple shallow neural network as a decoder to extract this information. In our benchmark tailored for Chinese music descriptions, we primarily evaluate the capabilities of music understanding models in terms of music description. We select widely used music understanding models as baselines and evaluate their performance on MuChin. Recent music understanding models include MERT-95M, MERT-330M [15], Jukebox-5B [16],

Music2Vec [17], and EnCodec [18]. Considering that Jukebox-5B is a pre-trained generative model not originally designed for music understanding, we use the method from [16] to encode audio through Jukebox-5B.

7.1.1 Evaluation Metrics

To evaluate the effectiveness of the music understanding models, we input music audio into these models and obtain their respective encoded sequences. Subsequently, for each model, we utilize an MLP containing an average pooling layer and 5 linear layers to extract 10 sets of descriptive music labels from the dimensions of its output encoded sequence. The pipeline of this process can be found in Figure 3.

- **Semantic Similarity Score.** The BGE model [19], as a general word vector embedding model, has shown impressive performance on various tasks. We utilize the bge-large-zh-v1.5 model to calculate the semantic similarity between the generated labels and the original labels.

For each set of test data, we can determine the semantic similarity between them by encoding the labels into embedding vectors using the BGE model and calculating the outer product of these embedding vectors. We then sequentially enumerate each generated label with the original labels, calculate their semantic similarity scores, and then take the average of all values as the score for a specific model.

7.1.2 Results

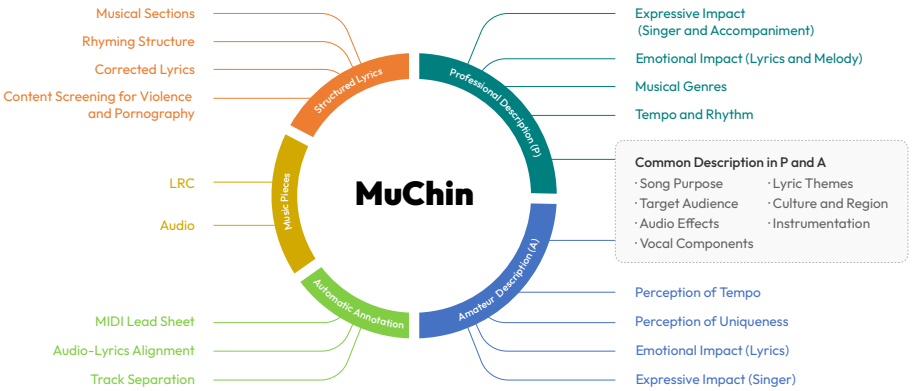


Fig. 2: The content structure of the MuChin corpus. The full names for the abbreviated descriptive dimensions presented in Table 10 are detailed in this figure.

Table 10 shows the semantic similarity scores of the five selected models. It can be observed that MERT, which encodes both audio and music attributes simultaneously, performs best in understanding and describing music. Thanks to its large number of parameters and training data, Jukebox also achieves good results, but its performance is not optimal due to the lack of emphasis on music attributes in its architecture. Furthermore, for the 95M and 330M versions of MERT, although their scores are not

Model		Jukebox	MERT-330M	MERT-95M	Music2Vec	EnCodec
Parameter Size		5B	330M	95M	95M	56M
Data (h)		60 ∼ 120k	160k	17k	1k	1k
	Average Score-P	0.5490(±0.1458)	0.5586(±0.1433)	**0.5640(±0.1425)**	0.5474(±0.1417)	0.4583(±0.1377)
	Tempo & Rhythm	0.4610(±0.1016)	**0.4650(±0.1013)**	0.4607(±0.0958)	0.4604(±0.1026)	0.4587(±0.1092)
	Emo. Impact (L & M)	0.5312(±0.0939)	0.5350(±0.0903)	**0.5396(±0.0857)**	0.5311(±0.0924)	0.4860(±0.0920)
	Cult. & Reg.	0.5166(±0.2107)	0.5340(±0.2139)	**0.5390(±0.2110)**	0.5120(±0.2094)	0.4072(±0.1261)
Professional	Vocal Components	0.5464(±0.1953)	0.5550(±0.1957)	**0.5713(±0.1989)**	0.5356(±0.1926)	0.4230(±0.1361)
Description	Song Purp.	0.5810(±0.2191)	0.5864(±0.2166)	**0.6040(±0.2230)**	0.5664(±0.2144)	0.4630(±0.1504)
	Mus. Genres	0.4600(±0.1239)	0.4644(±0.1172)	**0.4692(±0.1158)**	0.4610(±0.1207)	0.4297(±0.1219)
	Exp. Impact (S & A)	0.9146(±0.0541)	0.9280(±0.0476)	**0.9310(±0.0447)**	0.9190(±0.0576)	0.7085(±0.2888)
	Tgt. Aud.	0.4521(±0.1471)	0.4656(±0.1459)	**0.4683(±0.1417)**	0.4565(±0.1514)	0.3623(±0.0980)
	Instrum.	0.5083(±0.1647)	**0.5180(±0.1587)**	0.5156(±0.1592)	0.5063(±0.1727)	0.4043(±0.1426)
	Audio Eff.	0.5195(±0.1476)	0.5356(±0.1458)	**0.5425(±0.1483)**	0.5244(±0.1539)	0.4404(±0.1122)
	Average Score-A	0.5894(±0.1353)	0.5900(±0.1284)	**0.5923(±0.1284)**	0.5770(±0.1417)	0.4602(±0.1449)
	Perc. of Tempo	**0.4600(±0.1521)**	0.4540(±0.1475)	0.4580(±0.1456)	0.4463(±0.1407)	0.4065(±0.0994)
	Emo. Impact (L)	0.5977(±0.1780)	0.5894(±0.1798)	**0.6006(±0.1780)**	0.5806(±0.1827)	0.4430(±0.1320)
	Cult.& Reg.	0.4565(±0.1013)	0.4539(±0.0975)	**0.4575(±0.0949)**	0.4510(±0.1023)	0.4324(±0.0972)
Amateur	Vocal Components	**0.5195(±0.1208)**	0.5190(±0.1216)	0.5186(±0.1227)	0.5117(±0.12 00)	0.4795(±0.0950)
Description	Song Purp.	0.5240(±0.2377)	0.5210(±0.2356)	**0.5410(±0.2422)**	0.5201(±0.2428)	0.3801(±0.1532)
	Perc. of Uniq.	0.5356(±0.2076)	0.5356(±0.2115)	**0.5547(±0.2085)**	0.5060(±0.1942)	0.4130(±0.1191)
	Exp. Impact (S)	0.9404(±0.0328)	0.9385(±0.0315)	**0.9460(±0.0315)**	0.9297(±0.0477)	0.7144(±0.2640)
	Tgt. Aud.	0.4417(±0.1041)	0.4448(±0.1114)	**0.4530(±0.0951)**	0.4353(±0.1220)	0.3933(±0.1075)
	Instrum.	0.7144(±0.0737)	**0.7153(±0.0537)**	0.6787(±0.0333)	0.6807(±0.1059)	0.4219(±0.2092)
	Audio Eff.	0.7056(±0.1448)	**0.7275(±0.1465)**	0.7144(±0.1326)	0.7110(±0.1586)	0.5176(±0.1725)

Table 10: Evaluation results of selected music understanding models on the music audio understanding. The metrics of description presented in the table can be referenced to the **descriptive dimensions** of P and A on the right side of Figure 2. After encoding music by the models, we employ an MLP to output descriptive tags corresponding to these dimensions. The **pipeline** of this process can be found in Figure 3. The method for calculating the **semantic similarity** scores between the model's output results and the test set labels can be referenced in Section 7.1

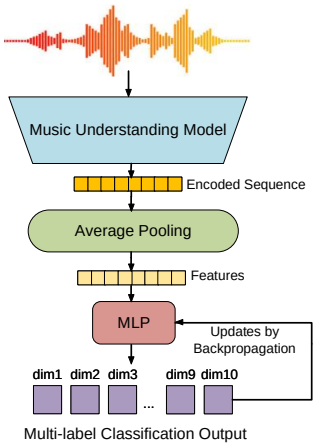
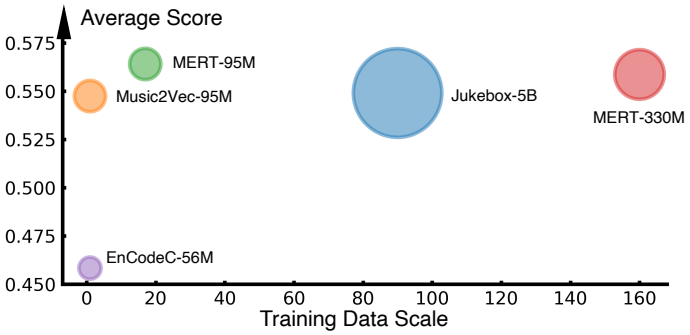
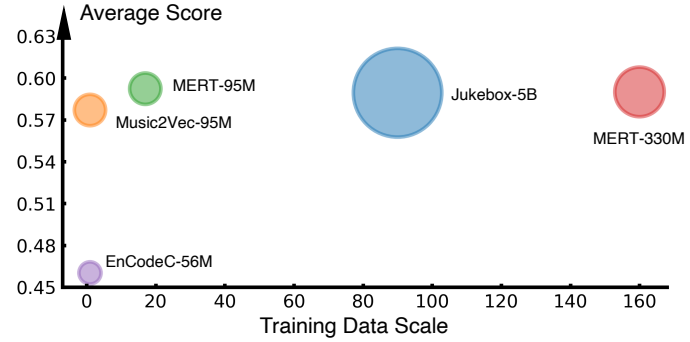


Fig. 3: The pipeline of evaluating music understanding models.

far apart, we observed an inverse-scaling effect in several dimensions, consistent with the phenomenon mentioned in the original MERT paper. Specifically, for objective music attributes such as rhythm and instrumentation, MERT-330M performs better, but for most descriptive dimensions with subjectivity, MERT-95M performs better. Therefore, we speculate that, as described in the original MERT paper, as the amount of data and parameters increases, MERT introduces more music attribute information, making it easier for the model to extract music attributes, but this may lead to the dilution of some information related to audio description. This also indicates that music attributes extracted through MIR cannot be directly used for music description benchmarks. Figure 4 shows, despite having fewer parameters and a smaller amount of training data, MERT-95M performs best overall in the task of professional and colloquial music description.



(a) Average Score - ***Professional***



(a) Average Score - ***Amateur***

Fig. 4: Evaluation of selected music understanding models on the benchmark as represented in a scatter plot.

8 Annotation Platform Interface Types, Screenshots, and Annotation Result Samples

This section will briefly introduce the platform.

- **Accounts and Login.** The platform employs an access control system, assigning a specific role to each user account. Users can log into their own accounts, view and complete assigned tasks, and then submit their results.
- **Annotation Interface.** After logging in and selecting a specific music clip, the annotator is directed to the annotation interface designed for the task. This interface includes a media player and a dedicated text box. Users can control the progress bar and playback speed of the media player. Additionally, the music description annotation interface integrates a comprehensive dictionary and search tools, allowing users to directly select appropriate descriptive terms from the dictionary or search for specific terms as needed.
- **Quality Assurance Interface.** After logging in and selecting a specific music clip, the quality assurance inspector is taken to the quality assurance interface. For Type A tasks, the inspector is responsible for evaluating the annotations provided by two users simultaneously. The interface displays these annotations side-by-side, highlighting differences for easy comparison. The inspector can then decide which annotation is correct, make adjustments to either, or choose to re-annotate the clip. For Type B tasks, the interface displays a single complete annotation for the inspector to verify and rate. The inspector simply reviews the annotation and submits a score.
- **Administrator Interface.** The administrator has the authority to view the submissions of any designated user, including annotators and quality assurance inspectors. Both the annotation and quality assurance interfaces include a feedback button for reporting platform issues, enabling annotators and quality assurance inspectors to communicate with the administrator to resolve problems.

We provide screenshots of several platform pages as examples, as shown in Figure 5 to Figure 9.

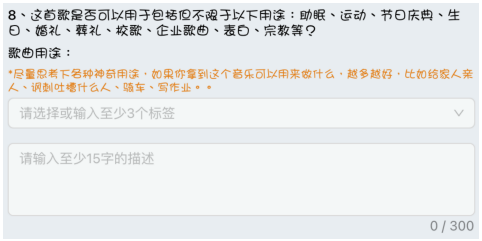


Fig. 5: Screenshot of the "Song Purpose" section during the description annotation phase.

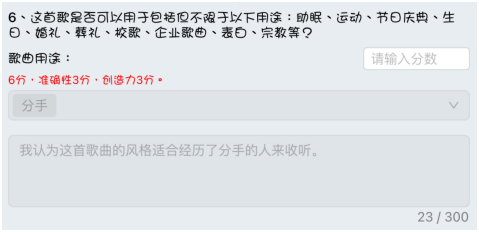


Fig. 6: Screenshot of the "Song Purpose" section during the quality assurance phase.

This section presents a series of annotation examples, including professional and colloquial music descriptions, as well as musical sections and rhyme schemes from MuChin, as shown in Figure 11 to Figure 15.

On one hand, we seamlessly integrate musical section annotations into the lyrics by marking the start of each section with a section label just before its lyrics begin. We use a string containing "c" and "R" markers to represent rhyming information:

Fig. 7: Screenshot of the "Instruments" section during the description annotation phase.

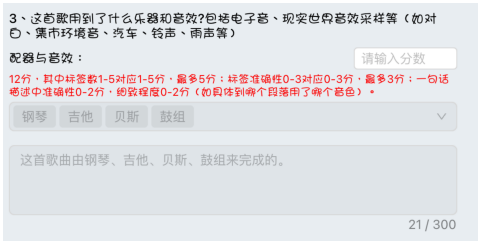


Fig. 8: Screenshot of the "Instruments" section during the quality assurance phase.

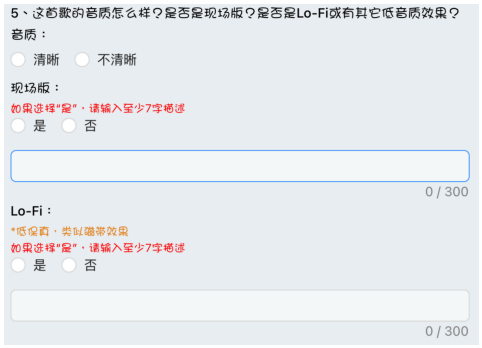


Fig. 9: Screenshot of the "Audio Effects" section during the description annotation phase.

an "R" is added at the end of any line that rhymes with the previous one, while "c" indicates a non-rhyming word. This method is used to compile all annotated lyric information—including the theme, sections, and rhyming details of the lyrics—into a JSON file.

On the other hand, during the phase dedicated to annotating music descriptions, we collect text descriptions for each piece of music from various perspectives. Each annotation consists of several descriptive terms and a comprehensive descriptive text. To enhance the richness of these descriptions, we integrate these terms into the descriptive text, which is then combined with the text. Furthermore, we concatenate descriptions from different aspects to create a single, detailed annotation that captures the multifaceted nature of the music.

9 Automatic Annotation and Data Preprocessing

9.1 Automatic Annotation Tasks

Current algorithms for annotating text descriptions, lyrics, and musical sections do not perform ideally as they rely on subjective human evaluation. However, other types

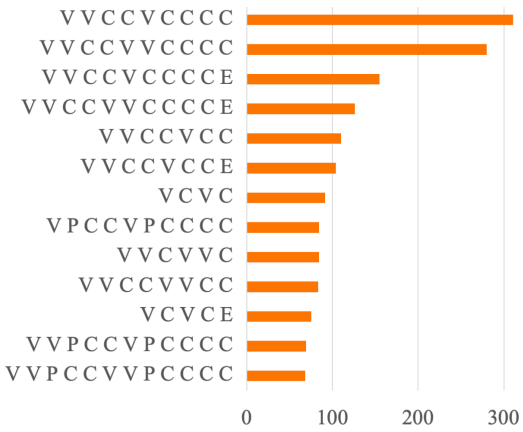


Fig. 10: Distribution of song structures. The interval labels on the left side of the histogram represent the various sections of a song. Specifically, "i" stands for "intro," "v" corresponds to "verse," "c" denotes "chorus," "p" indicates "pre-chorus," "b" means "bridge," and "e" represents "outro."

(verse1)
晚上来临了
ccccR
游戏通关了
ccccR
我们的爱情早已结束了
ccccccccR
(verse2)
我的心碎了
ccccR
你也解脱了
ccccR
你就像只船开走了
cccccccR
(chorus1)
船要起航了
ccccR
你要出发了
ccccR
到处漂泊的你也许会累了
ccccR

Fig. 11: Sample snippet of structural annotation.

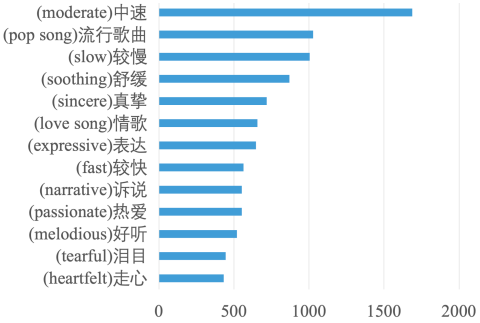


Fig. 12: Distribution of amateur descriptive labels

Fig. 13: Distribution of professional descriptive labels

of information, such as phoneme alignment, vocal separation, and audio-to-MIDI transcription, do not have a significant correlation with human perception. Furthermore, manually annotating these aspects is challenging and requires substantial effort and time. Currently, there is a plethora of mature algorithms capable of performing these tasks, which will be discussed in Appendix 9.2. Therefore, we employ data preprocessing algorithms to automatically annotate these contents without manual annotation or intervention, and directly integrate them into our dataset.

Main Question: 这首歌带给你的感受?
Main Question: How does this song make you feel?

Label Selection
Q1: 特色感受
Q1: Perception of Uniqueness
A1: "情歌","青春","积极面对","动听"
A1: "Love song," "Youth," "Face positively," "Melodious."
Q2: 快慢感受
Q2: Perception of Tempo
A2: "欢快","踩点","跟着哼唱"
A2: "Cheerful," "On beat," "Hum along."
Q3: 表现力感受（歌手）
Q3: Expressive Impact (Singer)
A3: "感悟","情深意切","余音袅袅","动情"
A3: "Insight," "Deep emotion," "Lingering sound," "Moving."
Q4: 情绪感受（歌词）
Q4: Emotional Impact (Lyrics)
A4: "成长","追忆","愉悦","释然"
A4: "Growth," "Reminiscence," "Joy," "Relief."

Compose Description
A: 这是一首正能量的歌曲，在成长中难免会遇到困难挫折，克服它们继续向前，向着人生的目标奔跑，要无所畏惧。
A: This is a positive song that acknowledges the inevitable difficulties and setbacks encountered during growth. It encourages overcoming these obstacles and continuing to move forward fearlessly towards the goals of life.

Fig. 14: Sample snippet of amateur description annotation.

9.2 Data Preprocessing

- **Music Genre Clustering.** To mitigate subjective bias and ensure a diverse range of descriptions for various music genres, it is crucial to assign a wide array of music genres to annotators, thereby enriching the diversity of annotations. To achieve this, we utilize MERT [15], a pre-trained music audio encoder, to process the audio data. Subsequently, we cluster the encoded data, yielding 1000 unique audio clusters. We then evenly distribute music data from these clusters, ensuring that annotators receive a balanced mix of music for annotation. This approach ensures that each music cluster is described by a range of annotators, thus significantly enhancing the diversity and richness of the annotated data.
- **Vocal and Track Separation.** To make the dataset suitable for tasks such as accompaniment generation, melody generation, and vocal synthesis, we apply Demucs [20, 21] for vocal separation, separating the vocals from the music accompaniment in the audio files. Additionally, considering the needs of a broader range of music-related tasks, we also separate individual instrument tracks, such as drums and bass.

Main Question: 这首歌带给你的感受?
Main Question: How does this song make you feel?

Label Selection
Q1: 表现力感受（歌手和伴奏）
Q1: Expressive Impact (Singer & Accompaniment)
A1: "美妙的声音","温暖磁性的男声","完美","柔和的男声"
A1: "Exquisite voice," "Warm and magnetic male voice," "Perfection," "Gentle male voice."
Q2: 情绪感受（歌词和旋律）
Q2: Emotional Impact (Lyrics & Melody)
A2: "感性的","放松和安慰","浪漫情感"
A2: "Sentimental," "Relaxing and comforting," "Romantic emotions."

Compose Description
A: 这首歌整体表现了较为感性的情感主题，描述了两人的陪伴和旅行方面的故事。歌手的表现力很棒，情感充沛，技法高超，很快能把人带入到一个舒缓平静的氛围中。
A: The song overall conveys a sentimentally charged emotional theme, depicting the story of companionship and travel between two individuals. The singer's performance is impressive, brimming with emotion and technical proficiency, swiftly drawing the listener into a soothing and tranquil atmosphere.

Fig. 15: Sample snippet of professional description annotation.

- **Phoneme-level Alignment in Audio-Lyrics.** To prepare audio-lyric pairs for applications such as vocal synthesis, it is necessary to align them at the phoneme level. We used the Montreal Forced Aligner (MFA) [22] for this task, initially achieving an accuracy of 67%. While MFA showed commendable 95% accuracy in aligning monophonic phonemes with single characters, its performance was degraded by inaccuracies in marking the offsets of melismatic phonemes. These phonemes are characterized by multiple pitches sung on a single syllable or note, which complicates the alignment process and reduces overall accuracy. To address this, we optimized the MFA algorithm, focusing on accurately identifying and aligning melismatic phonemes. Furthermore, we implemented features to recognize and annotate significant pauses and breaths during singing. These enhancements significantly improved our final alignment accuracy to 97%.
- **Automatic Pre-annotation.** To improve the efficiency of future manual annotation, we implemented specific software for automatic pre-annotation for certain tasks related to lyric annotation. For rhyming annotation in lyrics, we use a specialized program to pre-annotate the rhyme of each line. For theme annotation in lyrics, we employ a fine-tuned Qwen to preliminarily identify the main theme of each music's lyrics. During the formal annotation stage, these pre-annotations serve as a basis for manual review. Annotators can evaluate the accuracy of these automatic annotations and make adjustments as needed, or use them as a guide for their own annotation work.
- **Simplified Music Notation Transcription.** To facilitate the use of MIDI for tasks related to symbolic music, we transcribe the audio in MuChin into simplified music notation. These scores, a simplified form of MIDI notation, are created

using Sheet Sage [23], which utilizes the encoding model of Jukebox [24]. This conversion facilitates the application of MuChin to a wide range of tasks related to symbolic music.

10 Annotation Guidelines

The detailed annotation guidelines provided to human annotators are available online. The full documents can be accessed in both English and Chinese at the following links:

English Version:
https://docs.google.com/document/d/1_Ep7Sd1VH7NJxSTCxhWv0bGRXH7ox3BalVZ6Ci9v4io/edit?usp=sharing

Chinese Version (Original):
https://drive.google.com/file/d/10n6z7bpCEVmPA6opghLXCP8FjoC6tdrC/view?usp=share_link

References

[1] Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., He, X.: Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1316–1324 (2018)

[2] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research **21**(140), 1–67 (2020)

[3] Xue, L.: mt5: A massively multilingual pre-trained text-to-text transformer. arXiv preprint arXiv:2010.11934 (2020)

[4] Liu, Y.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 **364** (2019)

[5] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., et al.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)

[6] Dettmers, T., Pagnoni, A., Holtzman, A., Zettlemoyer, L.: Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems **36** (2024)

[7] Melechovsky, J., Guo, Z., Ghosal, D., Majumder, N., Herremans, D., Poria, S.: Mustango: Toward controllable text-to-music generation. arXiv preprint arXiv:2311.08355 (2023)

[8] Agostinelli, A., Denk, T.I., Borsos, Z., Engel, J., Verzetti, M., Caillon, A., Huang, Q., Jansen, A., Roberts, A., Tagliasacchi, M., et al.: Musiclm: Generating music from text. arXiv preprint arXiv:2301.11325 (2023)

[9] Bai, J., Bai, S., Chu, Y., Cui, Z., Dang, K., Deng, X., Fan, Y., Ge, W., Han, Y., Huang, F., Hui, B., Ji, L., Li, M., Lin, J., Lin, R., Liu, D., Liu, G., Lu, C., Lu, K., Ma, J., Men, R., Ren, X., Ren, X., Tan, C., Tan, S., Tu, J., Wang, P., Wang, S., Wang, W., Wu, S., Xu, B., Xu, J., Yang, A., Yang, H., Yang, J., Yang, S., Yao, Y., Yu, B., Yuan, H., Yuan, Z., Zhang, J., Zhang, X., Zhang, Y., Zhang, Z., Zhou, C., Zhou, J., Zhou, X., Zhu, T.: Qwen technical report. arXiv preprint arXiv:2309.16609 (2023)

[10] Baichuan: Baichuan 2: Open large-scale language models. arXiv preprint arXiv:2309.10305 (2023)

[11] Zeng, A., Liu, X., Du, Z., Wang, Z., Lai, H., Ding, M., Yang, Z., Xu, Y., Zheng, W., Xia, X., et al.: Glm-130b: An open bilingual pre-trained model. arXiv preprint arXiv:2210.02414 (2022)

[12] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., et al.: Gpt-4 technical report. arXiv preprint arXiv:2303.08774 (2023)

[13] Ratcliff, J.W., Metzener, D., *et al.*: Pattern matching: The gestalt approach. Dr. Dobb's Journal **13**(7), 46 (1988)

[14] Devlin, J., Chang, M.-W., Lee, K., Toutanova, K.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In: NAACL HLT 2019, vol. 1, pp. 4171–4186 (2019)

[15] Li, Y., Yuan, R., Zhang, G., Ma, Y., Chen, X., Yin, H., Lin, C., Ragni, A., Benetos, E., Gyenge, N., Dannenberg, R., Liu, R., Chen, W., Xia, G., Shi, Y., Huang, W., Guo, Y., Fu, J.: MERT: Acoustic Music Understanding Model with Large-Scale Self-supervised Training (2023)

[16] Castellon, R., Donahue, C., Liang, P.: Codified audio language modeling learns useful representations for music information retrieval. ISMIR (2021)

[17] Li, Y., Yuan, R., Zhang, G., Ma, Y., Lin, C., Chen, X., Ragni, A., Yin, H., Hu, Z., He, H., et al.: Map-music2vec: A simple and effective baseline for self-supervised music audio representation learning. ISMIR (2022)

[18] Défossez, A., Copet, J., Synnaeve, G., Adi, Y.: High fidelity neural audio compression. arXiv preprint arXiv:2210.13438 (2022)

[19] Xiao, S., Liu, Z., Zhang, P., Muennighoff, N.: C-Pack: Packaged Resources To Advance General Chinese Embedding (2023)

[20] Rouard, S., Massa, F., Défossez, A.: Hybrid transformers for music source separation. In: ICASSP 23 (2023)

[21] Défossez, A.: Hybrid spectrogram and waveform source separation. In: Proceedings of the ISMIR 2021 Workshop on Music Source Separation (2021)

[22] McAuliffe, M., Socolof, M., Mihuc, S., Wagner, M., Sonderegger, M.: Montreal Forced Aligner: Trainable Text-Speech Alignment Using Kaldi. In: Proc. Interspeech 2017, pp. 498–502 (2017). https://doi.org/10.21437/Interspeech.2017-1386

[23] Donahue, C., Liang, P.: Sheet sage: Lead sheets from music audio. Proc. ISMIR Late-Breaking and Demo (2021)

[24] Dhariwal, P., Jun, H., Payne, C., Kim, J.W., Radford, A., Sutskever, I.: Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341 (2020)