Diff-BGM: A Diffusion Model for Video Background Music Generation Supplementary Material

Sizhe Li[1] Yiming Qin[1] Minghang Zheng[1] Xin Jin[2,3] Yang Liu[1*]
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Beijing Electronic Science and Technology Institute
[3]Beijing Institute for General Artificial Intelligence
{lisizhe, minghang, yangliu}@pku.edu.cn kevinqym@stu.pku.edu.cn jinxinbesti@foxmail.com

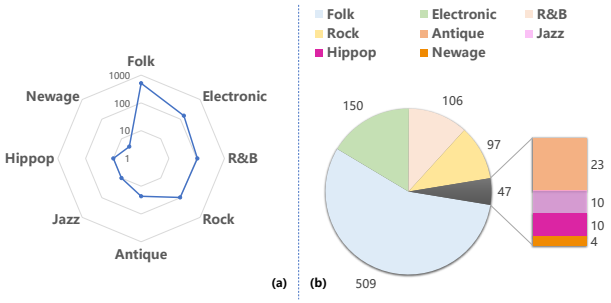


Figure 1. **Statistic of music style in BGM909.** We show the distribution of style types in the radar chart(a), and the amount of data and the diversity of styles in the pie chart(b).

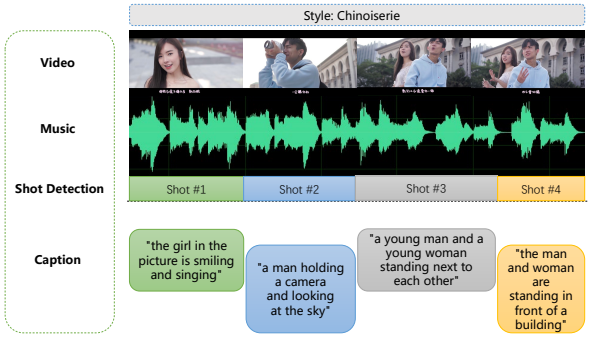


Figure 2. **Illustration of sample in BGM909.** A sample of BGM909 includes music-video pair, shot detection, corresponding caption, style, and other related metadata of MIDI.

*Corresponding author

1. Detailed Analysis of BGM909 dataset

1.1. Styles

As shown in Fig. 1, we divided the songs into 8 different styles, which shows the diversity of the music styles of BGM909. This diversity assists the model in learning different styles of background music. For instance, videos showcasing rural landscapes are more likely to be paired with 'Folk' background music. In this way, by providing styles of music, the intrinsic correspondence between video and music can be further learned based on the music diversity.

1.2. Data Samples

We give a sample of our annotation of a video-audio pair as shown in Fig. 2. For each sample, we provide the corresponding video-audio pair, shot detection results, captions and other metadata.

2. Additional Ablation Studies

We provide more ablation studies in this section, including the design of the feature selector and the shape of the mask in the segment-aware cross-attention layers.

2.1. Feature Selector

To verify the sequential formation of various aspects of music during different stages of generation, we employ distinct features to control music generation at different timesteps within the denoising process as shown in Tab. 1., which is aimed at attaining higher-quality music. In this set of experiments, we compare the generated music with the original music. The closer each indicator is to the original music, the higher the quality of the generated music. The first block of Tab. 1 gives the characters of the original music. In the second block, we consistently use features within the same modality throughout. Then in the third block, we

Methods	Music Quality		
	PCHE→	GPS→	SI→
Real(BGM909)	2.717	0.708	0.486
Only Video	2.835	0.514	0.396
Only Language	2.849	0.641	0.521
Random	2.568	**0.653**	0.425
Concat	2.870	0.634	0.528
Video+Language	2.840	0.626	0.536
Language+Video	**2.840**	0.601	**0.521**

Table 1. **Ablation studies on feature selector.** We use different strategies to select features as conditions between the given video and language features. The best results are indicated in bold.

combine features from two modalities with different methods. "Random" means that in each timestep, we randomly select features from one modality. "Concat" means that in each timestep, we concat the features from the two modalities. "Video+Language"(Language+Video) represents that, in the earlier timesteps, we initially use video (language) features, but as the generation process progresses, at a certain timestep, we switch to using language (video) features. The results show that when using two types of features, the quality of generated music is generally higher than only using features from one modality. According to our observations, the generative model produces music by first generating the melody followed by the rhythm. The musical melody is influenced by the video content, whereas language features provide better semantic information, making them suitable for guiding melody generation. On the other hand, the music rhythm is influenced by the dynamic information in the video; hence, frame-level video features are more suitable for guiding rhythm generation. Therefore, under the "Language+Video" condition, the quality of the generated music surpasses other feature selection methods.

2.2. Segment-Aware Cross-Attention layers

We change the mask shape k in the segment-aware cross-attention layers as shown in Tab. 2. k is used to generate the attention mask as follows:

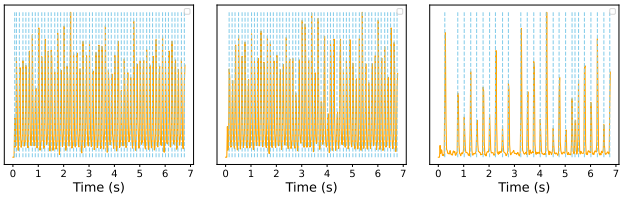
$$\mathcal{M}ask_{i,j} = \begin{cases} 1, & k \cdot \gamma \leq i, j < k \cdot (\gamma + 1) \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $\mathcal{M}ask \in \mathbb{R}^{T \times T}$ represents the attention mask, $\gamma \in \{0, 1, .., \frac{T}{k}\}$ represents the number of contexts.

We compare different window size k when generating attention masks. As shown in Tab. 2, when k increases, longer-term contexts interfere with music generation, introducing redundant information. Conversely, reducing k results in a smaller context size for the interaction between music and video features, negatively impacting video-music correspondence. Therefore, we select an appropriate feature shape and set k to 8. To select an appropriate context size, we also attempted to *dynamically* set the mask shape for each sample as shown in the last row of Tab. 2. Leveraging the shot detection information provided by BGM909, we calculate the duration of each shot within the samples and choose the mask shape closest to the changes in shots. However, since the duration of the samples we sampled is often much shorter than the shot durations, this also led to larger context sizes. Consequently, in Diff-BGM, we ultimately employed a fixed context size. We believe this is an issue worth further investigation.

3. Rhythm Visualization

We visualize the rhythm of the generated music during the denoising process. During the denoising process of Diff-BGM, the timestep ranges from 1000 to 0. We use language features as condition at the beginning stages of denoising where the timestep is from 1000 to 200, and use visual features at the subsequent stages where the timestep is from 200 to 0. We select rhythm visualization images at three different timesteps $t = 1000$, $t = 400$, $t = 0$ in Fig. 3. Each image represents one exact timestep. The $t = 1000$ image indicates the sampled noise and the $t = 400$ image shows the generated music after 600 denoising steps, which still contains some noise. The $t = 0$ image represents the final output music. For each image (timestep), the orange line represents the volume over time and the blue dotted lines represent the time when the accent occurs according to the volume change. If the dotted blue lines are evenly distributed and have a proper density, then we consider the music to be rhythmic. As shown in the first two figures of Fig. 3, the accents are disorganized, indicating a poor rhythm, and the last figure contains well-distributed accents, leading to a piece of rhythmic music. We can conclude that in the initial stages of generation, the rhythm is not prominent. However, in the subsequent stages, the rhythm gradually emerges during generation.



$t = 1000$ $t = 400$ $t = 0$

Figure 3. **Rhythm visualization at different timesteps during the denoising process.** The figure shows the rhythm changes when the timestep goes from 1000 to 0. The orange line represents the volume and the blue lines show music frames where accent occurs.

We also quantify the rhythm by proposing a new metric **Accent-Distance-Variance(ADV)** to visualize the effect of

Window Size	Music Quality			Video-Music Correspondence		
	PCHE→	GPS→	SI→	P@5↑	P@10↑	P@20↑
Real(BGM909)	2.717	0.708	0.486	—	—	—
k=32	**2.721**	0.789	0.523	13.83	23.27	41.23
k=16	2.792	**0.680**	0.540	12.63	23.58	40.89
k=8	2.840	0.601	**0.521**	13.28	23.91	**44.10**
k=4	2.767	0.604	0.531	**13.87**	**24.09**	42.25
Dynamic	2.729	0.775	0.534	13.07	23.80	41.44

Table 2. **Ablation studies on segment-aware cross-attention layers.** We use different window sizes k to measure the effect of our SAC-Att layers. Dynamic means we choose window size dynamically according to the sample.

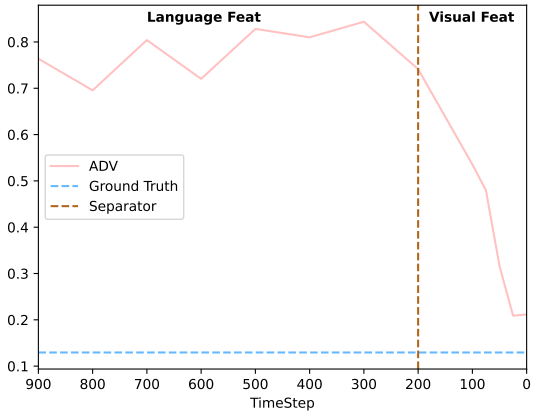


Figure 4. **Visualization of Accent-Distance-Variance(ADV) during the denoising process.** The timestep goes from 1000 to 0 and 13 timesteps are sampled during the whole denoising process. The ADV (pink) line represents the ADV value of music at each timestep. The Ground Truth line (blue) represents the ADV of the original music. The Separator line (brown) represents that when the timestep from 1000 to 200, we use the language feature for control and from 200 to 0 we use the visual feature. Small ADV value means rhythmic music.

different features used in different denoising steps and the formation of rhythm. We split a piece of music into n equal-length segments and detect the time locations of remarkable accents for each segment. After obtaining the time locations of accents, we calculate the distance between adjacent locations and conduct a normalization operation on the distances. Then values lower than a threshold are deleted due to different forms of rhythmic presentation. We then calculate the variance of this distance array. For dense accents always represent noise, we multiply the result by the number of detected accents and divide it by a factor which is the max distance of the distance array. The ADV is calculated as follows:

$$L_{\text{AVD}} = \frac{l \times var(normalize(D)_{deleted})}{d_{max}} \qquad (2)$$

where D is the distance array, l is the the number of detected accents and d_{max} means the max value of D. In order to restrict the value of ADV from 0 to 1, we use half sigmoid function and scale it by 2. The value of ADV at different timesteps is shown in Fig. 4, which indicates that when we use language features at the beginning stages of denoising, the rhythm is chaotic and when we use visual features in the subsequent stages, the rhythm becomes clear gradually.