1. Motion GPT and SM-GPT

We utilize the official implementation for our baseline method, Motion GPT [1]. We are using this implementation as the basis for our proposed modification, the SM-GPT. For our ablation study (see Table 3), we present the result obtained by retraining Motion GPT network using their original configurations.

2. Ablating Number of K

We first experiment by reducing the codebook and setting $K = 512$ to compensate for the lack of code variations when using two codebook (refer to Table 1). We perform a total of two experiments, one utilizing the same network structure as the dual codebook approach (using $D_1(x)$), while the other utilizes $D_2(x)$ with a slight modification at the decoder to use Conv1D instead of ConvTranspose1D. We found that the former performs better compared to the latter across all metrics, which indicates that the $D_2(x)$ is less ideal for the music decoder.

Afterwards, we experiment with the dual codebook design by changing the number of $K$ for both codebooks. To accomplish this, we add an FC layer to both $E(x)$ and $D(x)$ to adjust the dimension of the latent variable. This adjustment is necessary due to the downsampling effect, which results in a default latent variable dimension of 55 (8 times downsampling from 438-dim music features), which we utilize as the baseline value for $K$. The results from Table 1 indicate that this approach is less ideal than directly using the encoded 55-dim feature.

3. Codebook Analysis

We plotted the frequency distribution using three different values of $K$: 32, 55, and 64 (as shown in Figure 1, Figure 2, and Figure 3, respectively). We observe that setting $K$ as 55 or 32 encourages the network to fully utilize every key in the codebook, as there are less keys available to represent the data. However, we also notice that using $K = 32$ does not highlight the difference of each key as clearly as the other $K$ settings, as every key is utilized by multiple combinations of genre. This is not evident in the $K = 55$ or $K = 64$ setting, as some keys are used exclusively by specific genres.

4. Qualitative Result

Figure 4 some generated dance motion from our proposed method. We present a total of 6 sample of generated dance motions, each belonging to a different dance genre.

References


Table 1: M2C network codebook ablation study. The best and second best results are presented in bold and underline, respectively. We apply new norm during each of these training process.

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Accuracy ↓</th>
<th>Diversity ↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>GT (AIST++)</td>
<td>17.10 10.60</td>
<td>8.19 7.45</td>
</tr>
<tr>
<td>Single Codebook, $D_1(x)$</td>
<td>22.10 10.21</td>
<td>8.63 5.71</td>
</tr>
<tr>
<td>Single Codebook, $D_2(x)$</td>
<td>36.17 16.02</td>
<td>5.82 4.20</td>
</tr>
<tr>
<td>Dual Codebook, $K = 32$</td>
<td>20.86 8.84</td>
<td>6.61 6.02</td>
</tr>
<tr>
<td>Dual Codebook, $K = 48$</td>
<td>24.43 10.58</td>
<td>7.23 5.54</td>
</tr>
<tr>
<td>Dual Codebook, $K = 64$</td>
<td>26.60 10.03</td>
<td>5.93 5.50</td>
</tr>
<tr>
<td>M2C +SM-GPT +new norm</td>
<td>18.09 8.62</td>
<td>6.80 5.82</td>
</tr>
</tbody>
</table>
Figure 1: M2C’s music code distribution within each codebook ($K = 32$).
(a) M2C $C_1$ music code frequency distribution.

(b) M2C's $C_2$ music code frequency distribution.

Figure 2: M2C’s music code distribution within each codebook ($K = 55$).
Figure 3: M2C’s music code distribution within each codebook ($K = 64$).
(a) Dance motions generated from *Breakdance* music and dance motion seed.

(b) Dance motions generated from *Ballet Jazz* music and dance motion seed.

(c) Dance motions generated from *Hip Hop* music and dance motion seed.

(d) Dance motions generated from *Krump* music and dance motion seed.

(e) Dance motions generated from *Pop* music and dance motion seed.

(f) Dance motions generated from *Waack* music and dance motion seed.

Figure 4: Generated dance motions using our proposed method.