ABSTRACT
본 연구에서는 영상 배경 음악 생성 과제를 다룹니다. 전작들은 효과적인 음악세대를 이루지만 특정 영상에 맞춘 멜로디한 음악을 만들어내지 못하는 작품들도 있고, 비디오-뮤직의 리듬감 있는 일관성을 고려하는 작품은 하나도 없습니다. 주어진 영상과 일치하는 배경음악을 생성하기 위해, 먼저 영상과 배경음악의 리듬 관계를 설정해요. 특히 영상에서 나오는 타이밍, 동작 속도, 동작 민감도를 각각 비트, 시뮤 노트 밀도, 음악에서 나오는 시뮤 노트 강도와 연결합니다. 그런 다음 앞에서 언급한 리듬적 특징에 대한 로컬 제어와 음악 장르 및 악기에 대한 글로벌 제어를 가능하게 하는 제어 가능한 음악 트랜스포머인 CMT를 제안합니다. 객관적이고 주관적인 평가를 통해 생성된 배경음악이 입력 영상과 만족스러운 호환성을 달성함과 동시에 인상적인 음질을 달성했음을 알 수 있습니다.

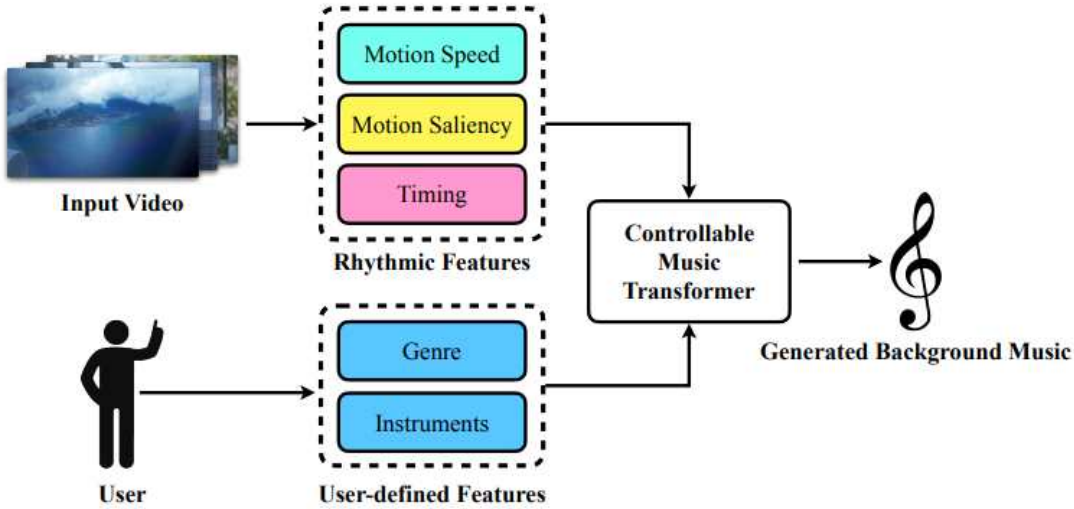


그림 1: 제안된 프레임워크 개요 우리는 영상과 음악의 세 가지 리듬 관계를 설정합니다. 추출된 리듬 기능은 사용자 정의 장르 및 악기와 함께 세심하게 설계된 제어 가능한 음악 트랜스포머로 전달되어 주어진 비디오에 적합한 배경 음악을 생성합니다.

1. INTRODUCTION

요즘 사람들은 소셜 비디오 공유 플랫폼에서 영상 편집 도구로 짧은 영상을 편리하게 편집하고 다른 사람들과 그들의 작품을 공유할 수 있습니다. 영상을 더 매력적으로 만들기 위해 배경음악을 추가하는 것은 흔한 일이지만, 음악이나 영화 편집에 대한 지식이 많지 않은 사람들에게는 그리 쉬운 일이 아니다. 많은 경우, 적절한 음악 자료를 찾고 동영상에 맞는 음악을 만들기 위해 조정하는 것은 이미 매우 어렵고 시간이 많이 걸립니다. 점점 더 광범위한 대중의 우려를 야기하고 있는 저작권 보호 문제는 말할 것도 없습니다. 따라서 주어진 비디오에 적합한 배경음악을 자동으로 생성하는 것은 현실 세계의 중요한 일이지만, 우리가 아는 한 멀티미디어 커뮤니티에서는 거의 연구되지 않고 있습니다. 딥러닝 모델을 기반으로 한 음악 생성을 다루는 이전 연구들이 있습니다 [27] [4] [10]. 그러나, 그 중 어느 것도 뮤직 비디오의 리듬 일관성을 고려하지 않았습니다. 본 논문에서 우리는 영상 배경 음악 생성 과제를 다룹니다. 교육 모델을 위해 쌍으로 구성된 비디

오 및 음악 샘플을 수집하는 등 비용이 많이 드는 데이터 중심 관행을 채택하는 대신, 비디오와 배경 음악 사이의 리듬 관계를 탐색하고 주석이 달린 교육 데이터에 의존하지 않고 변압기 기반 방법을 제안합니다. 그림 1에서는 이 작업에 대한 설명과 방법에 대한 개요를 제공합니다. 비디오는 다양한 패턴의 다양한 시각적 움직임을 포함할 수 있습니다. 예를 들어, 비디오 속의 한 남자가 빠르거나 천천히 걷고 있는데 갑자기 멈출 수도 있습니다. 이러한 장면에 적합한 배경음악을 생성하려면 동작의 속도와 변화를 고려해야 합니다. 특히 영상과 배경음악의 리듬 관계를 세 가지로 설정했어요. 첫째, 비디오 클립의 경우, 빠른 동작(예를 들어, 스포츠에서)은 강렬한 음악에 해당해야 하고, 그 반대도 마찬가지입니다. 따라서 동작 속도와 시뮬레이션 노트 밀도 사이의 양의 상관 관계를 구축합니다. 여기서 동작 속도는 평균 광학 흐름에 의해 계산된 작은 비디오 클립의 모션 크기를 나타내고 시뮬레이션 노트 밀도는 막대당 시뮬레이션 노트 수입니다. simu-note [25]는 그림 3에서와 같이 동시에 시작되는 노트 그룹입니다. 두 번째로, 샷 경계와 같은 독특한 움직임 변화는 강한 음악적 비트나 음악 경계에 부합되어야 하며, 관객이 시각과 청각의 영향을 동시에 느끼게 하여 보다 리드미컬한 비디오로 이어져야 합니다. 따라서 로컬-최대 모션 돌출성을 시뮬레이션 강도와 정렬합니다. 여기서 로컬-최대 모션 돌출성 레이블은 일부 리듬 키 프레임과 시뮬레이션 강도는 시뮬레이션 노트의 노트 수입니다. 세 번째, 영상과 생성된 배경음악 사이에 에필로그와 프롤로그를 동기화하는 것이 더 조화롭습니다. 즉, 배경음악은 영상의 시작과 끝과 함께 부드럽게 나타나고 사라져야 합니다. 따라서, 우리는 주어진 영상에서 타이밍 정보를 추출하고, 그것을 음악 생성 과정, 즉 비트 타임 인코딩을 안내하는 포지션 인코딩으로 받아들입니다. 우리는 합성어 [9] (CP)를 기반으로 음악 표현을 구축합니다. 우리는 이웃 토큰을 유형에 따라 그룹화하여 노트 관련 토큰과 리듬 관련 토큰을 위한 2-D 토큰을 구성합니다. 이러한 리듬감 있는 특징은 토큰에 추가 속성으로 추가됩니다. 또한, 우리는 주어진 영상에 맞춰 음악 생성 과정을 맞춤화하기 위해 그림 1의 하단과 같이 음악 장르와 악기를 이니셜 토큰으로 추가합니다. 선형 변압기[12]를 제안된 파이프라인의 백본으로 사용하여 주의력 계산에서 경량 및 선형 복잡성을 고려하여 음악 생성 프로세스를 모델링합니다. 교육 중에 Lakh Pianoroll Dataset(LPD)[4]을 사용하여 음악 모델링에 대한 모델을 교육합니다. 여기서 우리는 위의 음악적 특징을 직접 제공합니다. 추론을 위해, 시각적 특징은 비디오에서 얻어지고 생성 과정을 안내하는 데 사용됩니다. 요약하자면, 우리의 기여는 세 가지입니다. 1) 우리의 작업 비디오 배경 음악 생성을 위해, 우리는 비디오와 음악 사이의 몇 가지 주요 관계를 사용하지만 훈련 동안 쌍으로 구성된 비디오와 음악에 주석이 달린 데이터를 필요로 하지 않는 제어 가능한 음악 트랜스포머(CMT) 모델을 제안합니다. 2) 우리는 다음을 포함하여 음악의 새로운 표현을 도입합니다.음표 밀도와 시뮬레이션 노트의 강도를 통해 음악을 더 잘 생성하고 다중 트랙 생성 프로세스를 더 쉽게 제어할 수 있습니다. 3) 우리의 접근 방식은 성공적으로 음악을 비디오의 리듬과 분위기에 맞추는 동시에 높은 음악적 품질을 달성합니다. 우리는 보충 자료에 입력 영상과 생성된 음악의 데모 영상을 시연용으로 넣었습니다.

2. RELATED WORK

음악 표현입니다. 상징적인 음악 생성에 관한 대부분의 이전 작품들은 MIDI와 같은 이벤트 시퀀스로 표현된 음악을 입력으로 받아들입니다 [10] [16]. REMI [11]는 입력 데이터에 도량형 구조를 부과합니다. 즉, 막대, 비트, 코드 및 템포의 명시적 표시를 제공합니다. 이 새로운 표현은 국지적인 템포 변화의 유연성을 유지하는 데 도움을 주고 음악의 리듬과 조화 구조를 제어할 수 있는 기반을 제공합니다. 복합어 [9](CP)는 인접 토큰을 그룹화하여 REMI 토큰을 복합어 시퀀스로 변환하여 토큰 시퀀스의 길이를 크게 줄입니다. 본 논문에서 우리는 CP를 기반으로 한 표현을 사용합니다. 우리는 음악 토큰을 리듬 관련 토큰과 노트 관련 토큰으로 분류합니다. 우리는 음악의 전역 정보를 제공하기 위해 장르와 악기 유형을 초기 토큰으로 추가하고 생성 프로세스의 로컬 제어를 가능하게 하는 밀도 및 강도 속성을 추가합니다. 음악 세대 모델들이죠. 일부 최신 모델 [25] [23] [19]에서는 자동 인코더를 사용하여 기호 음악을 위한 잠재 공간을 학습하고 새 곡을 생성합니다. 일부 [27][4]에서는 피아노 롤을 2-D 영상으로 간주하고 컨볼루션 네트워크를 기반으로 모델을 제작합니다. 음악과 언어는 둘 다 시퀀스로 표현되기 때문에, 변압기와 변형이 음악 생성 모델의 백본으로도 자주 사용됩니다 [10] [11] [3] [9]. 기호 음악을 생성하는 것 외에도 일부 모델은 파형 [15] [5] [14]에서 직접 또는 전사 및 합성을 통해 간접적으로 오디오를 생성합니다[7]. 우리 모델은 선형 변압기[12]를 기반으로 하며, 시간 복잡성을 줄이기 위해 선형 주의 메커니즘을 구현합니다. 무성 비디오의 음악을 작곡합니다. 무성 비디오의 음악 작곡에 관한 이전 작품들은 바이올린, 피아노, 기타와 같은 다양한 악기를 연주하는 사람들이 포함된 비디오 클립으로부터 음악을 생성하는 데 초점을 맞추고 있습니다 [6] [21] [22]. 악기 종류라든지 리듬이라든지 세대결과의 많은 부분이 사람의 손의 움직임으로 직접 유추할 수 있기 때문에 음악이 어느 정도 결정되는 것입니다. 이와 비교하여, 우리의 방법은 일반 비디오에 적용되며 결정되지 않은 생성 결과를 생성하는 것을 목표로 합니다. 또한 현재 이 비디오 배경 음악 생성 작업을 위해 특별히 페어링된 임의 비디오 및 음악 데이터 세트가 없습니다. [1] [13]과 같은 일부 기존 시청각 데이터 세트에서는 비디오에 사람의 말소리와 같은 노이즈가 포함되거나 단순히 음악이 포함되지 않는 경우가 많습니다. 주석이 달린 훈련 데이터가 부족하기 때문에 시청각 데이터를 기반으로 한 기존의 지도 훈련 방법은 이 작업과 관련하여 작동하지 않습니다. 우리가 아는 한, 임의의 비디오 클립으로부터 음악을 생성하는 데 초점을 맞춘 기존 작업은 없습니다. 본 논문은 사용자 정의 음악 장르 및 악기와 함께 모션 민감성, 시각적 비트 및 영상의 전역 타이밍을 기반으로 배경 음악을 생성할 것을 제안합니다.

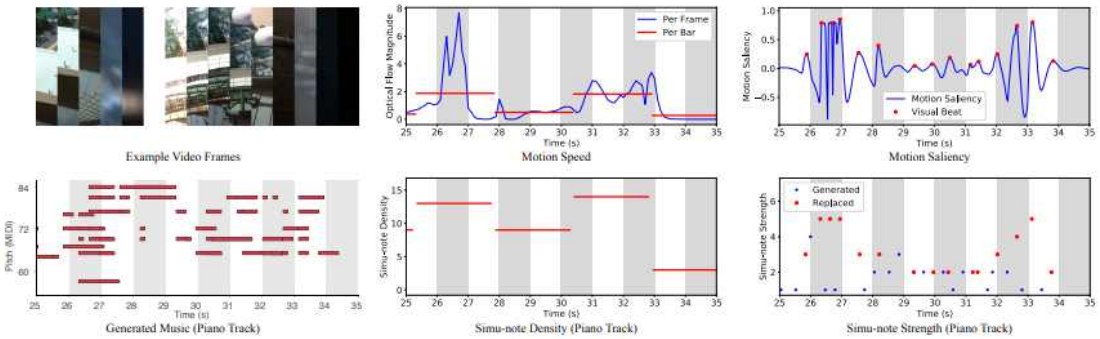


Figure 2: 우리의 방법을 사용하여 동영상과 배경음악의 리드미컬한 관계를 만들어냅니다. 여기에는 원본 비디오와 리듬 기능(상단), 생성된 음악 및 해당 기능(하단)이 나와 있습니다. 우리의 방법은 영상과 음악의 리드미컬한 관계를 구성하고 음악 생성을 안내하는 데 사용합니다.

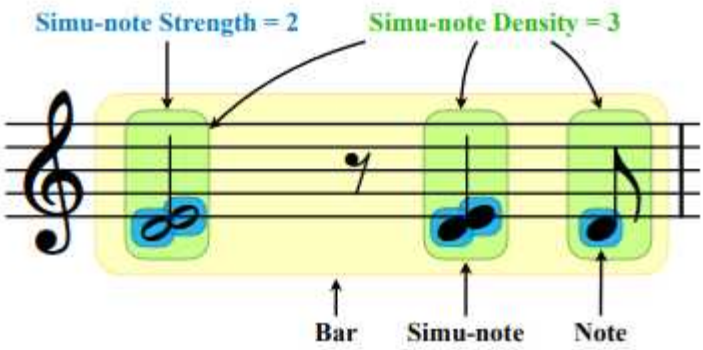


Figure 3: 시뮬레이션 밀도 및 강도에 대한 그림입니다.
시뮤-노트 밀도는 막대에 있는 시뮤-노트 수를 나타내며, 시뮤-노트 강도는 시뮤-노트의 노트 수를 나타냅니다.

3. 비디오-음악 리듬 관계 구축

로맨틱 영화를 볼 때 로맨틱한 음악을 듣거나 슈팅 게임 비디오로 강렬한 음악을 듣기를 기대할 수 있습니다. 리듬은 음악뿐만 아니라 영상에도 존재합니다. 그것은 비디오의 시각적 움직임이나 음악의 음표가 어떻게 일시적으로 분포되는지를 반영할 수 있습니다. 생성된 배경음악이 주어진 영상과 일치하도록 영상과 음악의 율동적인 관계를 분석하고 정립합니다. 아래 3.1절에서는 먼저 영상과 음악적 비트의 시간 사이의 연결을 구축합니다. 이러한 연결을 바탕으로 3.2절에서 우리는 운동 속도와 노트 밀도 사이의 관계를 정립합니다. 3.3항에서, 우리는 움직임의 민감성과 노트 강도의 관계를 구축합니다. 영상과 음악의 확립된 관계는 주어진 영상과 리드미컬하게 일치하는 배경음악의 생성을 안내하는 데 사용될 것입니다. 그림 2의 왼쪽에는 영상과 생성된 배경음악을 보여줍니다. 생성된 음악은 동작 속도가 높을 때 (그림 2의 중간에서 볼 수 있듯이) 큰 시뮬레이션 음의 밀도를 가지며 두드러진 시각적 비트가 발생할 때 (그림 2의 오른쪽에서 볼 수 있듯이) 큰 시뮬레이션 음의 강도를 가집니다.

3.1 Video timing and Music Beat

공식적으로, 우리는 T 프레임을 포함하는 비디오 ∈ RHX × X×3을 고려합니다. 우리는 다음 방정식을 통해 (번째(0 < t T T) 프레임을 비트 번호로 변환하는 것을 목표로 합니다:

$$f_{beat}(t) = \frac{Tempo \cdot t}{FPS \cdot 60}, \qquad (1)$$

여기서 tempo는 배경음악이 재생되어야 하는 속도이고, FPS는 영상의 본질적인 속성입니다. 우리는 1/4박자(one tick)를 가장 짧은 단위로 합니다. 또한, 우리는 역함수에 기초하여 ith비트를 비디오 프레임 번호로 변환할 수 있습니다.

$$f_{frame}(i) = f_{beat}^{-1}(i) = \frac{i \cdot FPS \cdot 60}{Tempo}. \qquad (2)$$

이 두 방정식은 영상과 음악 사이의 율동적인 관계를 구축하는 기본 블록 역할을 합니다.

3.2 Motion speed and Simu−note Density

먼저 전체 비디오를 다음과 같이 M 클립으로 나눕니다.

$$M = \left\lceil \frac{f_{beat}(T)}{S} \right\rceil, \qquad (3)$$

여기서 T는 비디오의 총 프레임 수이며 s = 4로 설정합니다. 이는 각 클립의 길이가 음악에서 4박자(1막대)에 해당함을 의미합니다. 그리고 나서 우리는 각 클립의 광학 흐름을 기반으로 운동 속도를 계산합니다.

광학 흐름은 비디오 동작을 분석하는 데 유용한 도구입니다. 공식적으로, 광학 흐름

$$f_t(x, y) \in \mathbb{R}^{\bar{H} \times \bar{W} \times 2}$$

는 두 개의 연속 프레임 사이의 개별 픽셀의 변위를 측정합니다.

$$I_t, I_{t+1} \in \mathbb{R}^{H \times W \times 3}.$$

거리 및 속도와 유사하게, 우리는 광학 흐름 크기 Ft를 t번째 프레임에서 운동 크기를 측정하기 위한 절대 광학 흐름의 평균으로 정의합니다.

$$F_t = \frac{\sum_{x,y} |f_t(x, y)|}{HW}, \qquad (4)$$

그리고 평균 광학 흐름 크기로서 (0 < m < = M) 비디오 클립의 운동 속도를 나타냅니다.

$$speed_m = \frac{\sum_{t=f_{frame}(S(m-1))+1}^{f_{frame}(Sm)} F_t}{f_{frame}(S)}. \qquad (5)$$

음악의 경우, 우리는 움직임 속도에 연결하기 위해 시뮬레이션 음의 밀도를 조작합니다. 여기서, 시뮬레이션 노트 [25]는 동일한 시작을 갖는 노트 그룹입니다.

$$simu\text{-}note_{i,j,k} = \{n_1, n_2, ..., n_N\}, \quad (6)$$

여기서 i는 ith 막대를 나타내고, j는 이 막대의 j번째 눈금(눈금 4개가 1비트), k는 악기를, n은 단일 음을 나타냅니다. 음표와 비교했을 때, 시무노트의 개념은 7번째 화음이든 9번째 화음이든 하나의 시무노트가 되기 때문에 리듬에 더 중점을 둡니다. 더욱이, 바는 비어 있지 않은 시뮬레이션 주석의 그룹으로 표현될 수 있습니다.

$$bar_{i,k} = \{simu\text{-}note_{i,j,k} | simu\text{-}note_{i,j,k} \neq \varnothing, j = 1, 2, ..., 16\}, \quad (7)$$

여기서 j = 1, 2, ..., 16을 16개의 눈금으로 나눕니다. 그러면 막대의 시뮬레이션 밀도는 다음과 같이 정의됩니다.

$$density_i = \left| \{ j | \exists k \in \mathbb{K}, simu\text{-}note_{i,j,k} \in bar_{i,k} \} \right|. \quad (8)$$

그런 다음, 우리는 일련의 비디오와 Lakh Pianoroll Dataset의 음악에서 속도 및 밀도의 분포를 통계적으로 분석합니다. 우리는 해당 밀도 수준의 퍼센트에 기초하여 속도m의 값 범위를 밀도의 클래스 수와 동일한 16개 클래스로 구분합니다. 예를 들어, 교육 세트에 밀도 = 16인 5% 막대가 있을 경우 상위 5% 속도를 밀도 = 16으로 분류합니다. m번째 비디오 클립은 ith bar와 길이가 같기 때문에, 우리는 밀도를 4.2항에서 논의한 바와 같이 관계를 구축하기 위해 추론 단계에서 기밀 속도계로 교체합니다.

3.3 Motion Salienct and Simu-note Strength

t번째 프레임에서의 모션 민감도는 두 연속 프레임 사이의 모든 방향의 광학 흐름의 평균 양의 변화로 계산됩니다. 그런 다음 로컬 최대 모션의 민감성과 거의 일정한 템포를 모두 가진 프레임을 선택하여 일련의 시각적 비트[2]를 얻습니다. 각 시각적 비트는 이진 튜플 (t, s)이며, 여기서 t는 프레임 지수이고 s는 그것의 돌출성입니다. 그림 2와 같이, s는 갑작스러운 가시적 변화가 발생했을 때 큰 값을 가질 것입니다. 그림 3에서 볼 수 있듯이 시뮬레이션 노트 강도를 노트 수로 정의합니다.

$$strength_{i,j,k} = |simu\text{-}note_{i,j,k}|. \quad (9)$$

직관적으로, 시뮤 음의 강도는 확장된 화음이나 하모니의 풍부함을 나타내며, 청중들에게 그것의 진행과 함께 리듬감 있는 느낌을 줍니다. 모의 음의 강도가 높을수록 청중이 더 많은 청각적 영향을 느낄 것입니다. 뚜렷한 시각적 움직임이 선명한 음악 비트로 표현되어 영상을 더욱 리드미컬하게 만들 수 있도록 시각적 비트 돌출성과 시뮬레이션 강도의 긍정적인 상관관계를 설정합니다.

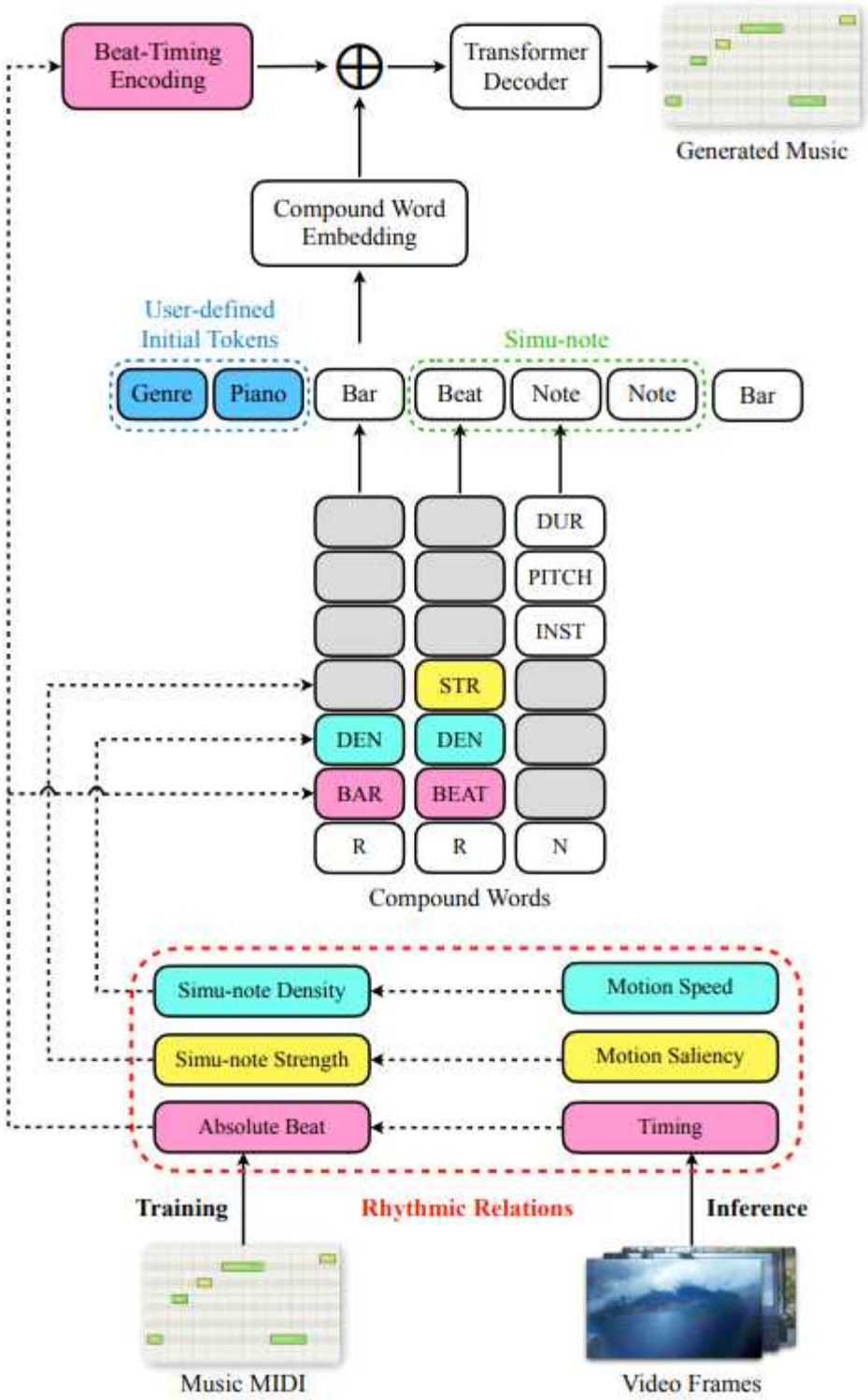


그림 4: 제안된 CMT 프레임워크의 그림입니다. 우리는 MIDI 음악(훈련 중) 또는 비디오(추론 중)에서 리듬감 있는 특징을 추출하고 음악 토큰의 표현으로 복합어를 구성합니다. 그런 다음 복합어 임베딩은 비트 타임 인코딩과 결합되고 시퀀스 모델링을 위해 변압기에 공급됩니다.

4. CONTROLLABLE MUSIC TRANSFORMER

위에서 확립된 비디오-음악 리듬 관계 외에도, 우리는 주어진 비디오에 대한 배경음악을 생성하기 위한 트랜스포머 기반 접근 방식인 제어 가능한 뮤직 트랜스포머(CMT)를 제안합니다. 전체적인 틀은 그림 4에 나와 있습니다. 위 절에 나와 있는 영상과 MIDI에서 리듬감 있는 특징을 추출합니다. 훈련 단계에서는 MIDI의 리듬감 있는 특징만 포함되어 있습니다. 추론 단계에서, 우리는 통제 가능한 음악 생성을 수행하기 위해 리듬적 특징을 비디오의 것으로 대체합니다.

4.1 Music Representation

제어 가능한 멀티트랙 상징적인 음악 생성을 위한 음악 토큰 표현을 설계합니다. PopMAG [18] 및 CWT [9]에서 영감을 받아 관련 속성을 단일 토큰으로 그룹화하여 시퀀스 길이를 단축합니다. 그림 4에서 볼 수 있듯이, 우리는 유형, 비트/바, 밀도, 강도, 악기, 피치 및 지속시간의 7가지 속성을 고려합니다. 이러한 속성을 비트, 밀도 및 강도를 포함한 리듬 관련 그룹(그림 4에서는 R로 표시)과 음이 속한 음조, 지속 시간 및 악기 유형을 포함한 음표 관련 그룹(그림 4에서는 N으로 표시)의 두 그룹으로 구분합니다. 그런 다음 유형 속성(그림 4의 R/N 행)을 사용하여 두 그룹을 구분합니다. 계산 모델링을 실현하기 위해, 우리는 그림 4에서 빈 속성과 같이 리듬 관련 토큰에서 노트 관련 속성을 없음으로 설정하고 그 반대의 속성을 설정합니다. 각 리듬 토큰에는 다음 노트 토큰 수를 나타내는 강도 속성이 있습니다. 또한 밀도 속성은 각 막대에서 단조롭게 감소하여 해당 막대에 남아 있는 시뮬레이션 노트 수를 나타냅니다. 각 토큰의 내장 벡터는 다음과 같이 계산됩니다.

$$p_{i,k} = \text{Embedding}_k\left(w_{i,k}\right), k = 1, ..., K, \qquad (10)$$

$$x_i = W_{in}\left[p_{i,1} \oplus ... \oplus p_{i,K}\right], . \qquad (11)$$

여기서 wi,k는 임베딩 전 ith 시간 단계에서 k 속성에 대한 입력 단어이고, K는 각 토큰의 속성 수입니다. 여기서 K = 7. Win은 연결된 내장 토큰을 Rd에 투영하는 선형 레이어입니다. 여기서 d = 512는 내장된 크기에 대해 미리 정의된 하이퍼 파라미터입니다. xi rd Rd는 내장된 토큰입니다. 또한 각 음악의 장르와 악기 유형을 초기 토큰으로 삼고 독립적인 임베딩 레이어를 적용합니다. 내장된 크기는 일반 토큰과 동일합니다.

4.2 Control

훈련을 마친 후, CMT 모델은 강도와 밀도 속성의 의미를 이미 이해했을 것으로 예상됩니다. 따라서, 우리는 주어진 영상과 음악을 더 조화롭게 만들기 위해 추론 단계에서 적절할 때에만 그 두 속성을 대체하면 됩니다. 밀도 치환입니다. 생성된 음악의 밀도가 비디오의 운동 속도 밀도와 일치하도록 각 막대 토큰의 밀도 속성을 광학 흐름에서 추출된 해당 비디오 밀도로 바꿉니다. CMT 모델은 이미 막대 토큰의 밀도 의미를 학습했으므로 이 막대에서 해당하는 수의 비트 토큰을 자동으로 생성하여 밀도를 제어할 수 있습니다. 강도 치환입니다. 마찬가지로, 우리는 생성된 시뮬레이션 노트의 강도를 제어하기 위해 비디오의 시각적 비트의 정보를 활용합니다. CMT 모델이 주어진 시각적 비트 이후 또는 그 비트 토큰을 예측한다면, 우리는 비트 토큰을 주어진 시각적 비트 및 강도로 교체합니다. 그런 다음 CMT 모델은 이 비트의 노트 토큰 수를 자동으로 예측하여 강도를 제어할 수 있게 합니다.

관리도에 대한 하이퍼 모수 C입니다. 우리는 또한 음악과 영상의 호환성과 멜로디 사이의 균형을 고려할 필요가 있습니다. 추론을 제약할수록 자연스럽지 않은 음악이 나오기 때문입니다. 이 문제를 해결하기 위해 생성된 음악에 대한 제어 정도를 나타내는 하이퍼 파라미터 C를 설계합니다. C가 클수록 추론에서 더 많은 제약이 추가될 것입니다. 즉, C가 0일 때 처음부터 음악을 얻고 C가 1일 때 완전한 호환성을 얻을 수 있습니다. 사용자는 자신의 필요에 따라 according를 설정할 수 있습니다. 비트 타임 인코딩입니다. 비디오의 시간 또는 길이 정보를 활용하기 위해 훈련과 추론을 모두 포함하는 토큰에 비트 타임 인코딩을 추가합니다. 이 설계는 CMT 모델에 프롤로그를 시작할 시기와 EOS 토큰을 예측할 시기를 알려줍니다. 비트 타임 인코딩은 주어진 비디오의 총 비트 수에 대한 현재 비트 수의 비율을 나타냅니다. 구체적으로, 우리는 그 비율의 값 범위를 폭이 같은 M bin(클래스)으로 나누고 학습 가능한 임베딩 레이어를 사용하여 토큰 임베딩과 동일한 차원으로 투영합니다. 그런 다음 이들을 합하여 CMT 모델의 입력을 형성합니다. i를 토큰의 인덱스, beati를 현재 단계에서 생성된 비트 수, Nbeat = fbeat(T)를 비디오의 총 비트 수, T를 비디오의 총 프레임 번호라고 가정하고 아래 방정식에 따라 비트-타이밍 인코딩을 계산합니다.

$$t_i = \text{Embedding}_t\left(round\left(M\frac{beat_i}{N_{beat}}\right)\right), \tag{12}$$

$$\vec{x_i} = x_i + BPE + t_i, \tag{13}$$

여기서 ti rd Rd는 토큰 i에 대한 비트 타이밍 인코딩, xi는 식 (11)에 설명된 임베딩 벡터, BPE는 식 (14) 및 (15)에 설명된 비트 기반 위치 인코딩입니다. 각 비디오를 100개의 빈으로 분리하도록 M = 100으로 설정합니다. xi->는 CMT 모델에 공급되는 최종 입력입니다. 또한 기존 위치 인코딩을 사용하는 대신 각 토큰에 대해 비트 기반 위치 인코딩을 도입합니다. 특히, 동일한 비트 내의 각 토큰은 동일한 위치 인코딩을 받습니다. 같은 비트의 여러 음이 순서와 상관없이 결국 동일한 오디오 세그먼트로 변환되기 때문에 음악 시퀀스의 의미 정보와 일맥상통합니다. i번째 비트 BPE의 비트 기반 위치 인코딩은 다음과 같이 계산됩니다.

$$BPE(beat_i, 2n) = sin(\frac{beat_i}{10000^{2n/d_{model}}}) \tag{14}$$

$$3PE(beat_i, 2n+1) = cos(\frac{beat_i}{10000^{2n/d_{model}}}) \tag{15}$$

여기서 dmodel = 512는 모형 은닉 크기이고 n [[0, ..., dmodel/2]는 d모델의 지수입니다. 비트 기반 위치 인코딩은 최종적으로 방정식 (13)의 각 내장 벡터 xi에 추가됩니다. 장르 및 악기 유형입니다. 우리의 방법에는 6가지 장르(컨트리, 댄스, 일렉트로닉, 메탈, 팝, 록)와 5가지 악기(드럼, 피아노, 기타, 베이스, 스트링)가 있습니다. 우리는 CMT 모델의 초기 토큰으로 그들로부터 각각 하나씩 가져갑니다. 사용자는 추론 단계에서 다른 초기 토큰을 사용하여 다양한 장르와 악기를 선택하여 영상의 감성에 맞는 배경음악을 생성할 수 있습니다. 위에서 언급한 제어 전략은 추론 단계에서 함께 결합될 것입니다. 더 자세한 추론 알고리즘은 알고리즘 1에 설명되어 있습니다.

4.3 Sequence Modeling

음악 토큰의 순서 (4.1절에서 설명)는 요소 간의 의존성을 모델링하기 위해 변압기 [24] 모델에 공급됩니다. 우리는 선형 변압기[12]를 등으로 사용합니다.

Model	Data	Without Control					With Control		
No.	-	1	2	3	4	5	6	7	8
Density	-	-	○	-	-	○	●	○	●
Strength	-	-	-	○	-	○	○	●	●
Beat-timing encoding	-	-	-	-	√	√	√	√	√
Pitch Histogram Entropy	4.452	3.634	2.998	3.667	3.573	3.617	3.496	4.044	**4.113**
Grooving Pattern Similarity	0.968	0.677	0.714	0.647	0.778	**0.810**	0.773	0.678	0.599
Structureness Indicator	0.488	0.219	0.227	0.215	0.223	0.241	**0.268**	0.211	0.200
Overall Rank ↓	-	5.000	5.000	5.333	4.000	**2.667**	3.667	4.667	5.667

표 1: 가락도에 대한 각 제어 속성의 객관적 평가 – 훈련 중 속성이 추가되지 않음을 의미합니다. ○ 속성은 훈련에서만 추가되고 추론 시간에는 제어되지 않음을 나타냅니다. ● 훈련 중에 추가하는 것뿐만 아니라 추론에서 주어진 비디오에서 상응하는 리듬적 특징을 가진 속성을 제어할 수 있다는 것을 의미합니다. √ 훈련 단계와 추론 단계 모두에서 비트 패리티 인코딩을 추가한다는 것을 나타냅니다. 자세한 내용은 5.3항을 참조하십시오.

Algorithm 1 Inference stage

```
Set initial genre and instrument tokens
repeat
  Predict next token with given beat-timing from the video based
  on sampling strategy
  if next token is bar then
    Replace its density attribute with prob of C
  end if
  if next token is after visual beat then
    Replace next token with visual beat and its strength with
    prob of C
  end if
  Append next token to music token list
until EOS token predicted
return music token list
```

주의력 계산에서 경량 및 선형 복잡성을 고려한 건축입니다. 다중 헤드 출력 모듈은 [9]의 설계에 따라 2단계 방식으로 각 음악 토큰의 7가지 속성을 예측합니다. 첫 번째 단계에서 모델은 변압기의 출력에 선형 투영을 적용하여 유형 토큰을 예측합니다. 두 번째 단계에서는 type을 사용하여 6개의 피드-포워드 헤드를 통과하여 나머지 6개의 속성을 동시에 예측합니다. 추론하는 동안, 우리는 생성된 토큰의 다양성을 증가시키기 위해 확률적 온도 제어 표본 추출 [8]을 채택합니다.

5. EXPERIMENTS

본 연구에서 제안하는 음악 생성에 대한 세 가지 제어 속성에 대한 절제 연구를 수행합니다. 둘 다 상대 평가가 실시된다 객관적. 객관적인 평가는 생성된 음악 자체의 품질에 초점을 맞춥니다. 여기서 각 비디오의 초기 토큰에 있는 모든 악기를 사용하여 각 장르에 대해 10개의 음악을 생성합니다. 그리고 나서 우리는 각 객관적인 메트릭에 대한 평균을 계산합니다. 주관적인 평가를 위해 설문지를 설계하고 사용자를 초청하여 생성된 음악의 품질과 해당 영상과의 호환성을 평가했습니다.

5.1 Dataset

우리는 CMT 모델을 교육하기 위해 Lakh Pianoroll Dataset(LPD)[4]을 채택합니다. LPD는 Lakh MIDI Dataset (LMD)에서 파생된 174,154개의 멀티 트랙 피아노 롤 모음집입니다 [17]. LPD의 LPD-5 클렌징 버전을 사용합니다. LPD는 클리닝 과정을 거치며 단일 MIDI 파일의 모든 트랙을 다섯 가지 공통 범주(드럼, 피아노, 기타, 베이스 및 스트링)로 병합합니다. 그런 다음 우리는 lpd-5-cleaned에서 3,038개의 MIDI 음악을 교육 세트로 선택합니다. 선택된 곡들은 태그트라움 장르 주석 [20]에서 여섯 가지 장르 (컨트리, 댄스, 일렉트로닉, 메탈, 팝, 록)로 나뉩니다.

5.2 Implementation Details

[9]의 설계에 따라 각 속성의 내장 크기를 어휘 크기, 즉 각각 (유형, 비트, 밀도, 강도, 피치, 지속 시간 및 계기) 속성에 대해 (32, 64, 64, 64, 512, 128, 32)를 기준으로 선택합니다. 이러한 내장 속성은 함께 연결되어 식 (11)의 숨겨진 크기를 모델링할 것으로 예상됩니다. 모델 세팅은 각각 8개의 어텐션 헤드가 있는 12개의 셀프 어텐션 레이어를 사용합니다. Feed-Forward 파트의 모델 히든사이즈와 내부 레이어사이즈는 각각 512, 2,048로 설정되어 있습니다. 각 계층의 드롭아웃 속도는 0.1로 설정됩니다. 입력 시퀀스 길이는 eEOS token 토큰으로 10,000까지 패딩됩니다. 초기 학습 속도를 1e-4로 설정하고 Adam을 최적화 도구로 사용합니다. 우리는 4개의 RTX 1080Ti GPU에서 약 28시간 동안 LPD 데이터 세트에 대해 100세기 동안 모델을 교육합니다. 목표 지표는 MusDr [26]로 계산됩니다.

5.3 Objective Evaluation

표 1의 제어 기능이 있는 각 모델에 대해 각 비디오에 대해 5개의 악기를 모두 사용하여 장르별로 10개의 MIDI를 생성합니다. 공정하게 비교하기 위해 각 모델에 대해 제어된 MIDI의 길이와 동일한 수의 MIDI를 제어되지 않고 생성합니다. 표 1에서는 먼저 LPD 데이터 세트에 대한 객관적 지표를 통계적으로 분석합니다. 생성된 음악은 채택된 메트릭과 관련하여 더 자연스러우도록 데이터 집합의 음악과 가까워야 합니다. 그런 다음 제안된 세 가지 리듬 기능이 없는 모델, 즉 비교를 위한 기준 모델로 사용되는 실험 No.1을 훈련합니다. 그리고 나서 우리는 각각의 리듬 특징을 하나씩 추가합니다. 즉, 표 1의 실험 번호 2, 3, 4. 밀도와 비트 타임 인코딩은 음악의 전반적인 구조를 개선하는 데 도움이 됩니다. 강도는 모델이 다양한 피치 클래스의 조합을 쉽게 학습하여 시뮬레이션 노트를 형성하여 피치 히스토그램 엔트로피를 높일 수 있도록 합니다. 실험 5번에 나온 것처럼 이 세 가지 제안된 리듬적 특징을 결합하면 모든 장점을 취해서 기준 모델보다 각 메트릭에서 높은 점수를 받아 전반적인 멜로디 향상이 향상되는 것을 알 수 있습니다. 그러나 우리가 주어진 영상, 즉

실험 번호 6, 7, 8로 그러한 속성들을 제어하려고 할 때, 우리는 음악 구조의 퇴화를 관찰합니다. 생성된 음악을 동영상의 리듬에 맞추도록 강제하기 때문에 합리적입니다. 따라서 5.4절에서는 구조 퇴화와 음악 비디오 호환성의 절충점을 찾기 위해 하이퍼 파라미터 C에 대한 사용자 연구를 수행합니다. 요약하자면, 실험 5번에 대한 가장 높은 전체 순위는 우리가 음악에서 추출한 리듬적 특징들이 추론을 가능하게 할 뿐만 아니라, 추출된 밀도와 강도는 CMT 모델이 리듬의 기본 패턴을 배우도록 촉진하기 때문에 생성된 음악의 전체적인 멜로디도 개선한다는 것을 보여줍니다. 배경음악으로. 다시 말해, 리듬감 있는 특징은 네트워크가 융합하는 것을 쉽게 만들어 음악의 구조를 개선합니다.

5.4 Subjective Evaluation

오늘날 음악 생성 모델을 평가하는 가장 좋은 방법은 사용자 연구를 사용하는 것입니다. 게다가, 객관적인 지표는 비디오와 음악이 일치하는 정도를 고려하지 않습니다. 이에 우리 모델에 대한 주관적 평가를 위한 설문지를 설계하고 36명을 참여시킵니다. 이 중 13명은 음악 관련 경험이나 음악 연주에 대한 기본적인 이해도가 있는 전문가로 평가받고 있습니다. 각 참가자는 하나의 입력 비디오에 해당하는 음악 몇 곡을 무작위로 듣고, 아래에 소개된 주관적 측정 기준에 따라 등급을 매기고, 자신의 선호도에 따라 순위를 매겨야 합니다. 음악이 길어질 수 있기 때문에 질문지는 완료하는 데 10분 정도 걸릴 수 있습니다. 우리는 우리 음악의 선율성을 평가하기 위해 몇 가지 주관적 지표[9]를 선택합니다. 1) 풍부함: 음악 다양성과 흥미성, 2) 정확성: 음표의 부재 또는 기타 연주 실수가 인식된 경우. (즉, 이상한 화음, 갑작스러운 침묵, 또는 악기의 어색한 사용) 3. 구조화: 주제를 반복하거나 음악적 아이디어를 발전시키는 것과 같은 구조적 패턴이 있는지 여부입니다. 게다가, 주어진 비디오와 음악의 호환성의 관점에서, 우리는 평가를 위해 다음 지표를 선택합니다: 1) 리듬감: 생성된 음악의 리듬이 비디오의 움직임과 얼마나 일치하는지. 예를 들어, 강렬한 스포츠 브이로그가 있습니다.

Model	Baseline	Matched	Ours
Melodiousness ↑	3.4	**4.0**	3.8
Compatibility ↑	3.4	3.7	**3.9**
Overall Rank ↓	2.3	1.9	**1.8**

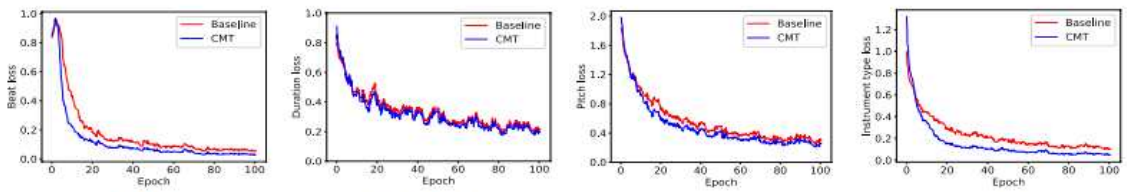
표 2: C = 0.7인 비디오와의 호환성과 멜로디에 대한 주관적 평가입니다. 우리의 음악은 특히 비디오와 호환될 때 훈련 세트의 일치된 데이터와 비교할 수 있는 성능에 도달합니다.

움직임이 큰 음악은 빠른 음악과 어울려야 합니다. 부드러운 움직임의 바지락과 부드러운 여행 브이로그는 느린 속도의 음악과 어울려야 합니다. 2) 대응: 생성된 음악의 주요 강세나 경계가 영상 경계나 시각 비트와 얼마나 일치하는지입니다. 예를 들어, 춤과 같은 리듬 동작과 몇몇 명백한 비디오 경계는 음악성을 향상시키기 위해 큰 스트레스를 동반해야 합니다. 3) 구조 일관성: 생성된 음악의 시작과 끝이 비디오의 시작과 일치해야 합니다. 마찬가지로, 음악과 비디오는 모두 프롤로그, 에필로그, 에피소드가 있어야 하며, 따라서 이러한 구조는 비디오를 더욱 조화롭게 만들기 위해 일치해야 합니다. 이러한 모든 주관적인 지표를 고려하여 생성된 배경음악에 대한 종합적인 평가를 하기 위해 참가자들에게 전반적인 품질에 기초하여 순위를 매기도록 요청하고, 그 다음 최종 결과로서 순위의 평균, 즉 전체 순위를 취합니다. 우리는 먼저 비디오와의 호환성과 생성된 음악의 멜로디 사이의 절충 문제

에 대한 적절한 값을 선택하기 위해 다양한 수준의 하이퍼 파라미터 Δ에 대해 실험합니다. 우리는 주어진 비디오를 선택하고, EACH Δ 값에 대해 추론 단계를 실행하고 하나의 뮤직 클립을 생성합니다. 모든 음악 클립은 참가자들에 의해 평가될 설문지에 포함됩니다. 그 결과는 표 3에 나와 있습니다. C = 1.0일 경우 비디오와 음악 간의 호환성이 향상되지만, 특히 정확도 측정 기준에서 음악의 선율을 저해하여 전체 순위가 낮아집니다. 리듬에 대한 제약으로 인해 모델이 상대적으로 부자연스러운 음을 생성하도록 강요되기 때문에 합리적입니다. 전체 순위를 고려하여 C = 0.7을 사전 정의된 초 매개 변수로 사용합니다. 그리고 나서, 우리는 우리의 음악을 멜로디함과 주어진 영상과의 호환성 측면에서 평가합니다. 기준선 모델은 제어 가능한 속성을 사용하지 않는 모델입니다. 또한 제안된 리듬 관계를 기반으로 훈련 세트의 음악과 비디오를 일치시키는 알고리즘을 설계합니다. 특히 비디오와 음악 작품이 주어지면 이들의 일치 점수(MS)를 다음과 같이 계산합니다.

$$MS(\mathrm{d}^{\mathrm{m}}, \mathrm{d}^{\mathrm{v}}, \mathrm{s}^{\mathrm{m}}, \mathrm{s}^{\mathrm{v}}) = \frac{1}{MSE(\mathrm{d}^{\mathrm{m}}, \mathrm{d}^{\mathrm{v}}) + MSE(\mathrm{s}^{\mathrm{m}} \odot \mathbb{1}(\mathrm{s}^{\mathrm{v}}), \mathrm{s}^{\mathrm{v}})}. \tag{16}$$

여기서 dm과 dv는 음악과 비디오에서 추출된 simu-note 밀도이며 동일한 크기로 잘립니다. 마찬가지로, sm과 sv는 잘라내기를 통해 simu-note 강도를 추출합니다. MSE는 평균 제곱 오차이며, ⊙는 하다마르 곱을 나타냅니다. 1(·)은 각 양수를 1로, 비양수 원소는 0으로 매핑합니다. 그런 다음 비디오 스타일에 따라 상위 5개 일치 음악 중에서 수동으로 선택합니다.



(a) barbeat loss (b) duration loss (c) pitch loss (d) instrument loss

그림 5: 기준선 모델과 CMT 모델의 손실 곡선입니다. 파란색 라인은 기본 모델이고 빨간색 라인은 우리가 제안한 CMT 모델입니다. 우리의 방법은 밀도, 강도 및 비트 타임 인코딩의 도움으로 향상된 수렴 속도를 보여줍니다. 더 나은 시연을 위해 a = 0.9로 각 손실에 대해 지수 평활을 수행합니다.

Metrics		C			
		0.0	0.3	0.7	1.0
Melodiousness	Richness ↑	3.6	3.4	**3.8**	3.7
	Correctness ↑	3.2	**3.7**	**3.7**	2.8
	Structuredness ↑	**3.6**	**3.6**	**3.6**	3.3
Compatibility	Rhythmicity ↑	3.2	3.5	**3.7**	**3.7**
	Correspondence ↑	2.6	3.3	3.7	**4.1**
	Structure Consistency ↑	2.9	**3.9**	3.8	3.8
Overall rank ↓		3.1	2.2	**2.1**	2.6

표 3: 선율 및 비디오와의 호환성에 대한 다양한 C 값에 대한 주관적 절제 연구입니다.우리는 높은 C는 높은 호환성으로 이어지는 반면, 낮은 C는 더 나은 멜로디로 이어지는 것을 관찰했습니다. C를 0.7로 설정하면 전체 성능이 피크에 도달합니다.

Attribute	Density	Strength	Time
Control Error	0.107	0.001	0.028

표 4: 밀도, 강도 및 시간 제어에 대한 오류율입니다. 우리의 방법은 음악의 세 가지 리듬감을 조절하는데 있어 인상적인 성능을 보여줍니다.

서로 다른 범주(편집, 미편집, 애니메이션 비디오)에서 3개의 비디오 클립을 선택하고 생성된 음악, 기준선 및 일치된 음악을 설문지에 제공합니다. 이러한 조합은 무작위로 섞이기 때문에 참가자들은 모델에 의해 생성된 것과 데이터 집합에서 선택된 것을 알 수 없습니다. Tab. 2는 우리가 생성한 배경음악이 전체적으로 매칭된 음악을 능가한다는 것을 보여줍니다. 일치하는 음악은 우리의 기준보다 더 나은 호환성을 보여주며, 우리가 제안한 리듬 관계가 영상의 전반적인 음악성에 가치 있고 유익하다는 것을 나타냅니다. 저희 작곡 음악의 멜로디함이 아직 트레이닝 세트에서는 진짜에 못 미치지만, 뛰어난 궁합이 그런 단점을 보완해 인간이 만든 음악보다 더 적합한 배경음악입니다.

5.5 Controlling Accuracy

우리는 제안된 세 가지 제어 가능한 속성의 정확도를 평가합니다. 우리는 음악에서 그 세 가지 속성을 다시 계산하고 비디오의 리듬적 특징과 우리의 음악 사이의 L2 거리를 컨트롤 에러로 잡습니다. 그런 다음 밀도, 강도 및 시간의 오류는 각각 막대당 평균 시뮬레이션 노트 수, 시뮬레이션당 평균 노트 수 및 총 비디오 시간으로 정규화되어 오류율을 형성합니다. 표 4에서 보듯이, 우리의 결과는 인상적입니다. 음악 밀도에 대한 제어 오류는 약 0.1인 반면 막대당 평균 시뮬레이션 노트 수는 9.9개입니다. 이는 각 막대당 비트 수가 주어진 비디오 밀도에 따라 대략 한 비트만 변동한다는 것을 의미합니다. 강도 제어 오류는 대부분의 시뮬레이션 노트가 주어진 비디오 강도와 정확히 같은 수의 노트를 가지고 있음을 보여줍니다.

5.6 Visualization

우리는 기준 모델과 CMT 모델 모두에서 리듬 관련 속성과 노트 관련 속성에 대한 손실 곡선을 시각화합니다. 결과는 그림 5에 나와 있습니다. 우리의 CMT 모델은 각 속성, 특히 비트 속성에 대해 더 빠른 수렴 프로세스를 가지고 있습니다. 즉, 우리의 추출된 리듬 기능은 모델이 음악에 대한 중요한 지식을 쉽게 파악할 수 있도록 하여 생성된 음악을 더 많이 불러올 수 있게 합니다.

6. CONCLUSION

본 논문에서는 비디오 배경 음악 생성이라는 미개척 과제를 다룹니다. 우리는 먼저 영상과 배경음악의 리듬 관계를 세 가지로 정립합니다. 그런 다음 음악 생성 프로세스의 로컬 및 글로벌 제어를 달성하기 위해 제어 가능한 음악 트랜스포머(CMT)를 제안합니다. 우리가 제안한 방법은 주어진 비디오와 멜로디하고 호환되는 음악을 생성하는 동안 훈련을 위해 쌍으로 구성된 비디오 및 음악 데이터를 필요로 하지 않습니다. 미래 연구에는 시각과 음악 사이의 보다 추상적인 연결(예: 감정과 스타일)을 탐색하고, 파형의 음악을 활용하고, 짝을 이룬 데이터에서 감독되지 않은 시청각 표현 학습을 채택하는 것이 포함될 수 있습니다.