國立臺灣大學電機資訊學院電信工程學研究所

碩士論文

Graduate Institute of Communication Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Master's Thesis

基於 Transformer 模型鋼琴伴奏風格轉換

Transformer-Based Piano Accompaniment Style Transfer

艾芯

Hsin Ai

指導教授: 楊奕軒 博士

Advisor: Yi-Hsuan Yang, Ph.D.

中華民國 114 年 6 月

June, 2025

doi:10.6342/NTU202501267

國立臺灣大學碩士學位論文
口試委員會審定書

MASTER'S THESIS ACCEPTANCE CERTIFICATE
NATIONAL TAIWAN UNIVERSITY

基於 Transformer 模型鋼琴伴奏風格轉換

Transformer-Based Piano Accompaniment Style Transfer

本論文係 艾芯 (姓名) R12942156 (學號) 在國立臺灣大學電信工程學研究所完成之碩士學位論文，於民國 114 年 6 月 18 日承下列考試委員審查通過及口試及格，特此證明。

The undersigned, appointed by the Graduate Institute of Communication Engineering on 18 (date) 6 (month) 2025 (year) have examined a Master's Thesis entitled above presented by Hsin Ai (name) R12942156 (student ID) candidate and hereby certify that it is worthy of acceptance.

口試委員 Oral examination committee:

楊奕軒 鄭皓中 蘇黎

(指導教授 Advisor)

所長 Director: 張宏亮

doi:10.6342/NTU202501267

Acknowledgements

首先，我要感謝我的指導教授－楊奕軒教授，在我碩士生涯中一路的引領和指導。從一開始我什麼都還不懂時，帶領我進入音樂 AI 這個領域，一步步耐心的教導我如何讀 paper、如何做研究，無私的給予我許多寶貴的建議和想法，啟發我對研究的興趣和熱忱。經過這兩年，我對於應用所學於自己喜愛的音樂上（特別是鋼琴），感到越來越有興趣，未來若有機會，想要繼續於這個領域深造。另外，也很感謝教授給予我們很大程度的研究自由，不僅讓我們自由選擇想做的題目，也經常關心我們的日常生活，並給予未來方向的幫助。特別感謝教授讓我在碩班期間同時兼顧召會生活，成為我這兩年莫大的扶持、加力和供應。

我也很感謝 Music AI Lab 的的每一位成員，很喜歡這裡的氛圍，不論是博班學長、同學、還是學弟妹，經常一起討論研究、互相幫助。特別感謝同樣做 symbolic 研究的博班學長，兩年來耐心的指導我們這些剛入門的碩士生，陪我們參加每一次的 small group meeting，提供我許多研究的想法和思路，每一次的討論都獲益良多，也激發我最後論文題目的靈感。

最後，謝謝我的家人，在我求學過程中始終無條件的支持我，讓我無後顧之憂的安心做研究，還有召會中每一位聖徒、同伴的照顧、扶持與代禱。感謝主，這兩年的碩士生涯真是滿了喜樂、感謝與讚美！

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

摘要

針對特定編曲家風格的流行鋼琴演奏版（piano cover）進行風格轉換，是符號化音樂生成領域中的一項獨特挑戰，其核心在於實現穩健的內容與風格解耦。本研究中，我們將「風格」定義為特定編曲家的伴奏模式——例如其特有的節奏密度 (rhythmic intensity)、複音織度 (polyphony) 、音域（pitch range）等伴奏型態；而將「內容」定義為核心的旋律及和聲。此任務的一項關鍵困難在於，即使是旋律本身也可能包含了編曲家的風格變化。本論文旨在解決此問題，我們確立了以導引譜 (lead sheet)——一種包含旋律與和弦進行的樂譜——作為「內容」的穩固基礎。透過提供一個明確的核心音樂結構，譜面得以有效去除鋼琴演奏中所附加的風格變化，為風格轉換提供了更清晰的分離基礎。在此之上，本研究系統性地比較了數種基於 Transformer 的架構，以探究直接基於 token (token-based) 的控制方法與更複雜的基於嵌入 (embedding-based) 策略的成效。值得注意的是，本研究框架的運作無需成對資料。我們的綜合評估顯示，儘管所有實現的方法都能有效捕捉目標編曲家的特徵，基於 token 的模型卻是一個更簡潔且有效的解決方案。它在風格轉換任務的兩大核心層面——內容保留與風格匹配——的客觀與主觀評估中，均取得了更優越的表現。這個關鍵發現提供了有力的實證證據：對於此類任務，利用導引譜來清晰地表示內容，能讓一個簡單的、基於 token 的模型實現風格轉換，為未來的研究提供了一個實際且有效的基準。

關鍵字：音樂風格轉換、鋼琴伴奏、內容－風格解耦、導引譜、Transformer

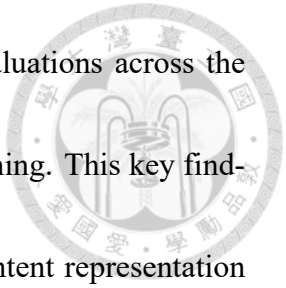
doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

Abstract

Arranger-specific style transfer for pop piano covers presents a unique challenge in achieving robust content-style disentanglement. For this work, we define arranger-specific style by unique accompaniment patterns, such as characteristic rhythmic intensity, polyphony, and pitch range. Content, conversely, is identified as the core melody and harmony. A key difficulty is that even performed melodies can contain an arranger's stylistic variations. This research addresses this by establishing the lead sheet as a robust anchor to decouple the musical content from stylistic variations, enabling a cleaner separation of style. Building on this foundation, we propose a Transformer-based framework to systematically compare the efficacy of a direct token-based conditioning approach versus more complex embedding-based strategies. Notably, this framework operates without requiring paired data. Our comprehensive evaluations demonstrate that while all implemented approaches successfully transfer the target arranger's characteristics, the simpler token-based model consistently proves to be a more effective and efficient solution. It

doi:10.6342/NTU202501267



achieved superior performance in both objective and subjective evaluations across the two core dimensions of the task: content preservation and style matching. This key finding highlights a crucial insight: leveraging a lead sheet for clear content representation allows a simple token-based model to achieve highly effective style transfer, providing a practical and efficient benchmark for future work.

Keywords: Music Style Transfer, Piano Accompaniment, Content-Style Disentanglement, Lead Sheet, Transformer

doi:10.6342/NTU202501267

Contents

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

List of Figures

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

List of Tables

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

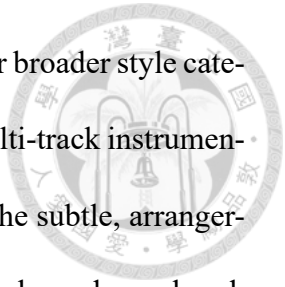
Chapter 1 Introduction

Computational music creation, a vibrant fusion of technology and artistry, empowers machines to compose or transform music, amplifying human creativity. For decades, symbolic formats like MIDI have been a cornerstone of this field, encoding musical events with a precision that offers a structured alternative to audio's complex waveforms [3]. This clarity makes MIDI ideal for tasks requiring fine-grained control, with research splitting into two broad categories: from-scratch generation and style transfer.

Our work focuses on style transfer within the specific domain of pop piano covers. These are instrumental arrangements where different arrangers artistically reinterpret a familiar melody using a wide variety of distinct accompaniment styles—defined by unique rhythms, textures, and harmonic choices. This phenomenon, where a consistent melody (content) is naturally paired with varied accompaniment patterns (style), makes piano covers a compelling case for computational modeling, particularly for the core challenge of content-style disentanglement.

While style transfer in symbolic music is a broad field of research encompassing diverse tasks such as genre adaptation [28], compositional rearrangement [23], expressive performance modification [39], timbre alteration [29], and emotion-driven generation [21], a significant gap remains when applying these concepts to the nuanced domain of

doi:10.6342/NTU202501267



piano cover arrangements. Most existing studies tend to focus on either broader style categories like genre transformation or the orchestrational challenge of multi-track instrumentation [9]. In contrast, the specific task of modeling and transferring the subtle, arranger-specific styles found within single-track piano MIDI remains relatively underexplored. This thesis aims to address this gap by developing and evaluating novel deep learning frameworks designed to achieve effective content-style disentanglement for this specific task, enabling the generation of accompaniments that reflect distinct arranger identities while preserving the original melodic and harmonic content.

Early music generation relied on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for melody creation or harmonization, but these models struggled with long-range dependencies, producing fragmented outputs [38], [14]. The advent of Transformers [35], with their self-attention mechanism, revolutionized sequence modeling by capturing global dependencies, making them ideal for MIDI's sequential nature . Transformers excel in generating coherent music over extended sequences, as demonstrated in tasks like multi-track composition [23] and pop piano performance [22]. Our interest lies in Transformer-based content-style disentanglement, where content (melody, preserved in lead sheets) is separated from style (arranger-specific accompaniment patterns).

Lead sheets, which encode the core melody and chords of a song, serve as a robust content anchor in our proposed framework, inspired by their successful use in prior work [36]. By explicitly defining the musical content, they naturally isolate the melody from the accompaniment, providing a clear structural foundation for style manipulation that is absent in entangled full performance representations. Grounded on this principle, our work investigates the most effective way to represent this lead sheet content and the target style

doi:10.6342/NTU202501267

for Transformer models. We explore and compare two primary conditioning strategies: a token-based approach, where both content and style are represented as discrete tokens, and embedding-based approaches, which utilize learned, continuous embeddings to represent either the lead sheet content or the target style.

Our work addresses the identified gap in arranger-specific style transfer by synthesizing key insights from two highly relevant and influential models: MuseMorphose's Transformer-VAE framework, which excels in fine-grained style modeling, and Compose & Embellish's lead sheet-based generation, which prioritizes structural clarity [37], [36]. Building on these foundations, this thesis presents a systematic investigation into how content and style are represented for this specific task. To answer the central research question of whether token-based or embedding-based strategies are more effective, we implement and conduct a comparative analysis of three distinct Transformer-based architectures, each designed to test a different representation approach.

Our results, validated on pop music datasets, demonstrate that a straightforward token-based method can deliver simple yet effective arranger-specific style transfer, outperforming the more complex embedding-based variants in our experiments. This key finding suggests that for transferring between a known, finite set of styles, a strong content representation (via lead sheets) combined with direct token-based conditioning is a highly efficient and potent strategy, potentially obviating the need for complex latent space modeling.

However, we also acknowledge the primary limitation of this token-based approach: its inability to perform zero-shot style transfer to unseen styles, as it relies on style-specific tokens learned during training. For such advanced tasks, we posit that the principles be-

doi:10.6342/NTU202501267

hind the embedding-based methods, which learn a continuous representation of style, hold greater potential and represent a valuable direction for future exploration.

The thesis is structured as follows: Chapter 2 reviews related work. Chapter 3 details the methodology. Chapter 4 outlines the experimental setup and evaluation metrics. Chapter 5 presents and discusses the results. Finally, Chapter 6 concludes the thesis and suggests directions for future work.

doi:10.6342/NTU202501267

Chapter 2 Related Work

This chapter surveys literature on symbolic music generation and style transfer, focusing on methods and representations relevant to arranger-specific piano accompaniment style transfer in single-track MIDI for pop piano covers. We explore music generation, style transfer techniques, content and style representations, and research gaps addressed by our work.

2.1 Music Generation

Symbolic music generation, leveraging MIDI's precise encoding of musical events (notes, durations, velocities), has been revolutionized by deep learning, which processes discrete tokens to create music automatically [3]. While non-deep learning methods hold value, neural networks dominate recent advances, particularly for piano-focused tasks. Early efforts used recurrent neural networks (RNNs), with Todd [33] pioneering monophonic melody generation. However, RNNs struggled with long-range dependencies due to vanishing gradients, producing fragmented outputs [33], [38]. To address this, long short-term memory (LSTM) networks enabled better sequence memory, as demonstrated by Eck and Schmidhuber [12], who generated blues improvisations with coherent rhythm and structure.

doi:10.6342/NTU202501267

Subsequent models improved polyphonic generation. Boulanger-Lewandowski et al. introduced RNN-RBM [2], outperforming traditional models but still limited in capturing long-term structure. The advent of deep generative models marked significant progress. MusicVAE [31], a hierarchical VAE, captured polyphonic music's long-term structure with strong interpolation and reconstruction capabilities. Generative adversarial networks (GANs) also emerged, with MidiNet [38] generating melodies bar-by-bar using a conditional mechanism based on chords, and MuseGAN [11] creating multi-track polyphonic music, showcasing GANs' versatility. Transformers [35], with their self-attention mechanism, excelled at long-sequence modeling, as seen in the Music Transformer [20] and Pop Music Transformer [22], which generated coherent piano performances and pop music, respectively .

Despite these advances, music's hierarchical nature—where melodies, harmonies, and rhythms combine—poses challenges. Two-stage frameworks simplify generation by first creating lead sheets (melody and chords) before stylizing them into performances. Compose & Embellish [36] exemplifies this, using lead sheets as content anchors to ensure structural clarity. Emotion-driven harmonization [21] adapts lead sheets for specific moods, highlighting their flexibility. Commercial tools like Band-in-a-Box and Google's Magenta project further demonstrate lead sheets' practical utility in generating user-driven music [3]. These insights inspired our token-based style transfer approach, leveraging lead sheets tokens as content with Transformers for arranger-specific piano covers.

doi:10.6342/NTU202501267

2.2 Music Style Transfer

Style transfer modifies musical attributes while preserving content, with style being a defining characteristic learned from datasets or model differentiation [8]. We define style as arranger-specific accompaniment patterns (non-melody notes, e.g., rhythms, chord textures) in pop piano covers, with content as the melody, encoded in lead sheets. Few studies address arranger-specific style transfer in single-track piano MIDI, a niche our work targets.

Early models like RNNs and CNNs, limited by local focus, gave way to generative adversarial networks (GANs) [13] and variational autoencoders (VAEs) [25]. CycleGAN [5] enables unpaired genre transfer in symbolic music, producing rich outputs without note limits. MIDI-VAE [4], a VAE-based model, disentangles style (pitches, dynamics, instrumentation) in multi-instrument polyphonic music using a shared latent space with a style classifier, achieving unaligned style transfer for complete works. Transformers [35] advanced style transfer, with MuseMorphose [37] using a Transformer-VAE for bar-level fine-grained transfer, allowing user-specified attributes like rhythmic intensity and polyphony. We explored embedding-based approaches, inspired by MuseMorphose [37], encoding lead sheet embeddings for content or full performance embeddings for style. Choi et al. [7] used a transformer autoencoder to derive global style representations from performances, combining them with temporal embeddings to control melody and style, also informing our embedding experiments.

Style transfer tasks vary: genre transfer alters broad structures [9], [5], compositional rearrangement [40] modifies harmony, expressive performance adjusts dynamics [39], and emotion-driven generation targets mood [21]. Homophonic style transfer, as in

doi:10.6342/NTU202501267

Lu and Su [27], focuses on accompaniment, adapting DeepBach [14] with WaveNet [34] and Gibbs sampling to transfer scores into Bach or jazz styles. Chord-based methods, like Groove2Groove [9], use synthetic data but focus on genre transfer by modifying the underlying harmonic structure. Content-style disentanglement is challenging due to unaligned data, necessitating unsupervised learning. VAE-based methods and GANs encode style into latent vectors but require complex tuning [32], [18], [30], [19]. Vector-quantized VAEs (VQ-VAEs) offer discrete control for timbre or rhythm transfer [10], [29]. These complexities motivated our token-based approach, detailed in Section 2.3.

2.3 Content and Style Representations

Style transfer hinges on effective representations. Token-based approaches, like REMI sequences, encode musical events (e.g., note-on, duration) as discrete tokens, offering precise control, especially for lead sheets, which compactly capture melody and chords [36], [21]. Embedding-based approaches encode features into continuous latent vectors via VAEs, enabling nuanced adjustments in rhythm or dynamics, as in MuseMorphose [37] or MIDI-VAE [4]. Lead sheets are widely used as content anchors in tasks like stylized accompaniments [36], phrase arrangements [40], and emotion-driven generation [21] .

Our contribution uses lead sheets as a melody-preserving bridge, facilitating content-style disentanglement without complex embeddings. Our token-based approach, leveraging lead sheets and style tokens, achieves effective arranger-specific style transfer, inspired by Compose & Embellish's [36] two-stage framework. We also investigated embedding-based methods, encoding lead sheet embeddings for content or full performance embeddings for style, but found the token-based method simpler and more effective, validated

doi:10.6342/NTU202501267

on pop music datasets.

2.4 Research Gaps

Research on arranger-specific style transfer in single-track piano MIDI for pop covers is limited, with most studies targeting broader tasks like genre transfer or multi-track modeling [4]. Transformer-based methods rarely use lead sheets for style transfer, and VAE-based models often prioritize complex latent spaces over structural clarity [37], [30]. Our work addresses these gaps by combining MuseMorphose's Transformer-VAE framework with Compose & Embellish's lead sheet-based approach, demonstrating that token-based leads and style tokens offer a simple, effective solution for arranger-specific style transfer, as explored in Chapter 3.

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

Chapter 3 Methodology

This chapter outlines the methodology for a Transformer-based model for piano accompaniment style transfer in popular music covers. The approach focuses on disentangling content (melody and harmony) from style (rhythmic patterns, textural elements, and decorative notes) to generate piano accompaniments in varied styles while preserving the song's core musical structure. The core concept integrates the lead sheet representation from Compose & Embellish [36] with the encoder-decoder Transformer VAE architecture from MuseMorphose [37] to achieve effective content-style disentanglement.

3.1 Content and Style Disentanglement

In piano covers of popular songs, the accompaniment provides harmonic and rhythmic support for the main melody, which is not a direct input but shapes the arrangement. Style variations across arrangers manifest in:

- Rhythm: Variations in note density, complexity, and syncopation.
- Harmony: Differences in chord selection, voicings (note arrangements within chords) and progression sequences.
- Texture: The degree of polyphony (simultaneous notes) and balance between sparse and full arrangements.

doi:10.6342/NTU202501267

- Decorative Notes: Embellishments or counter-melodies distinct from the main melody.

In this study, "accompaniment" refers to all piano notes excluding the main melody, flexibly played by either hand. The primary challenge is to transform the accompaniment's style while retaining the song's core content (melody and harmony). Disentanglement of content and style is critical, enabling independent style manipulation. A lead sheet, comprising the main melody and chord progression, serves as the content representation, abstracting stylistic details like rhythm and embellishments. This approach, inspired by Compose & Embellish, provides a clear foundation for style transfer by separating essential musical structure from performative elements.

3.2 Model Architecture

This study proposes three Transformer-based models, combining the lead sheet concept from Compose & Embellish [36] with the encoder-decoder Transformer VAE framework from MuseMorphose [37]. Each model processes lead sheets as content and incorporates style conditioning to generate stylistically varied accompaniments. Detailed configurations are provided in Section 3.5.1 and Table A.2.

3.2.1 Model 1: Decoder-Only with Token-Based Content and Style

Model 1 uses a decoder-only Transformer, adapted from Compose & Embellish [36], to generate a full performance (X) conditioned on a lead sheet (M) containing melody and chord information. The input sequence interleaves one-bar segments of lead sheet tokens (M) and performance tokens (X) (e.g., M(1), X(1), M(2), X(2)), with special tokens [Track_M] and [Track_X] marking each segment (see Figure 3.4 for sequence structure).

doi:10.6342/NTU202501267

A style token at the sequence's start guides the model to produce X in the target arranger's style. The model generates tokens autoregressively, minimizing negative log-likelihood with a causal attention mask. Ablation studies test alternative style token placements (e.g., per bar) and lead sheets without chords. Figure 3.1 illustrates the architectures.

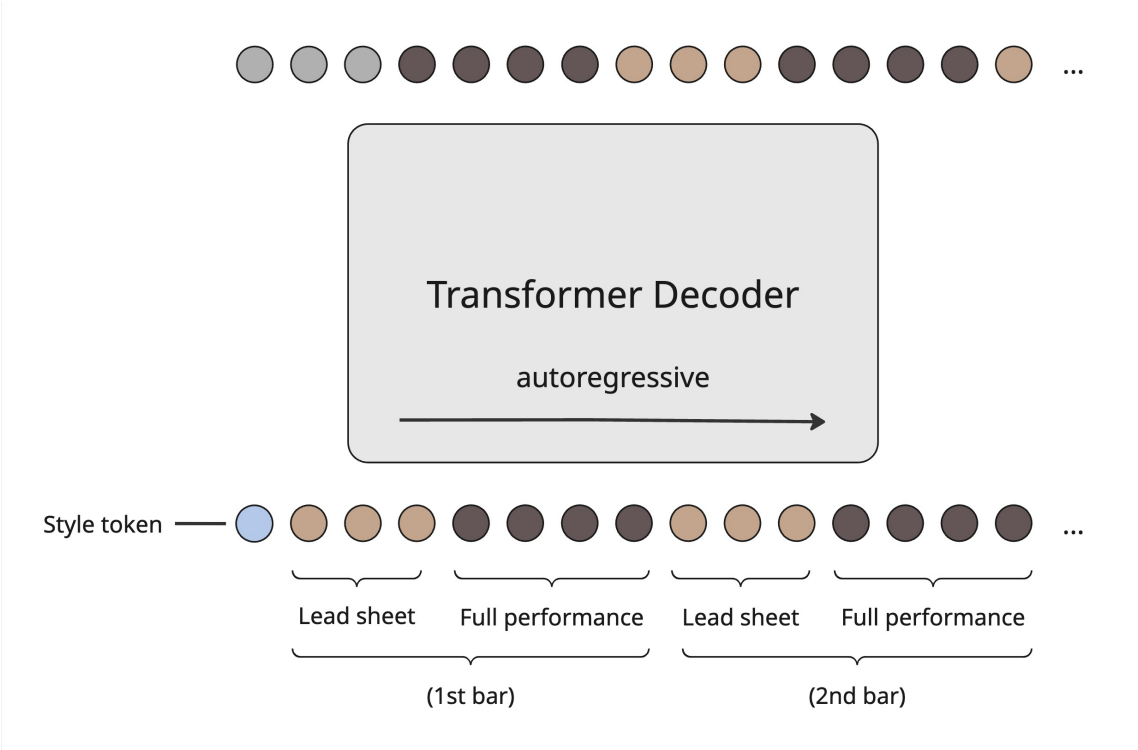


Figure 3.1: Model 1: Decoder-Only Transformer (content: lead sheet tokens, style: Arranger_ID token)

3.2.2 Model 2: Encoder-Decoder with Embedding-Based Content and Token-Based Style

Model 2 employs an encoder-decoder Transformer architecture. The encoder processes the lead sheet token sequence to generate contextualized content embeddings. The decoder generates the performance autoregressively, starting with a style token and using cross-attention to integrate content embeddings, ensuring alignment with the lead sheet and specified style. The lead sheet and performance are processed as 8-bar segments, as shown in Figure 3.2.

doi:10.6342/NTU202501267

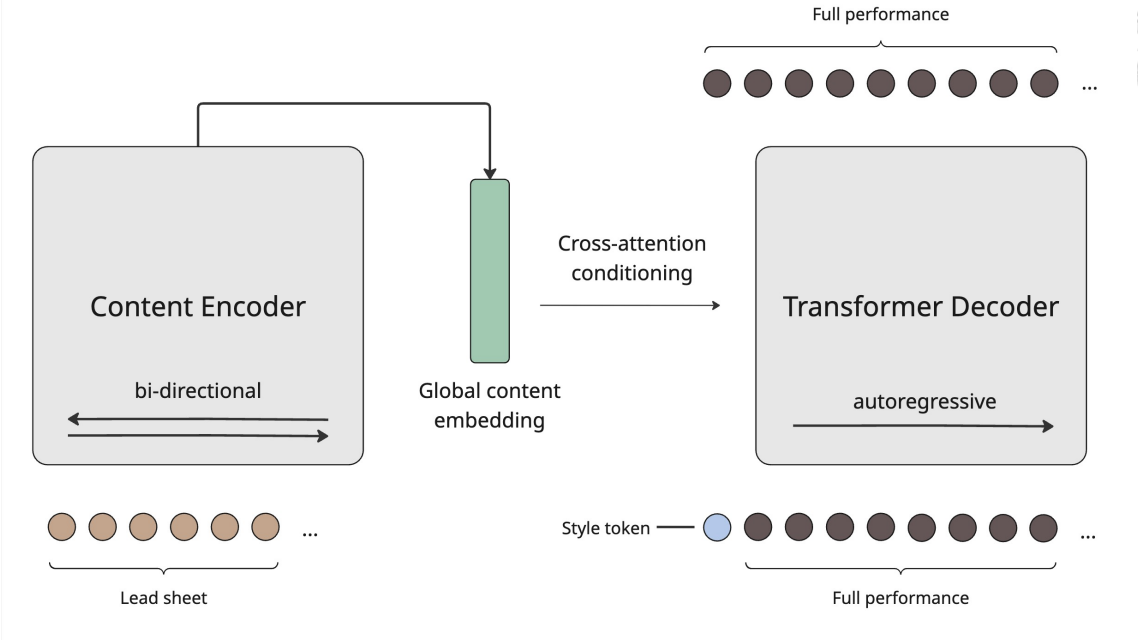


Figure 3.2: Model 2: Encoder-Decoder Transformer (content: lead sheet embedding, style: Arranger_ID token)

3.2.3 Model 3: Encoder-Decoder with Token-Based Content and Embedding-Based Style

Model 3 combines a style encoder and a main decoder, leveraging a $\beta - VAE$ [15] framework to enhance content-style disentanglement. The style encoder processes a 8-bar reference performance token sequence to produce a global style embedding. The main decoder uses an interleaved lead sheet and performance token sequence, similar to Model 1, with the style embedding injected via an in-attention mechanism [37]. The VAE bottleneck ensures robust separation of content (lead sheet) and accompaniment style. Ablation studies compare using identical versus adjacent 8-bar reference segments for style encoding. Figure 3.3 illustrates the sequence structure and style embedding integration.

doi:10.6342/NTU202501267

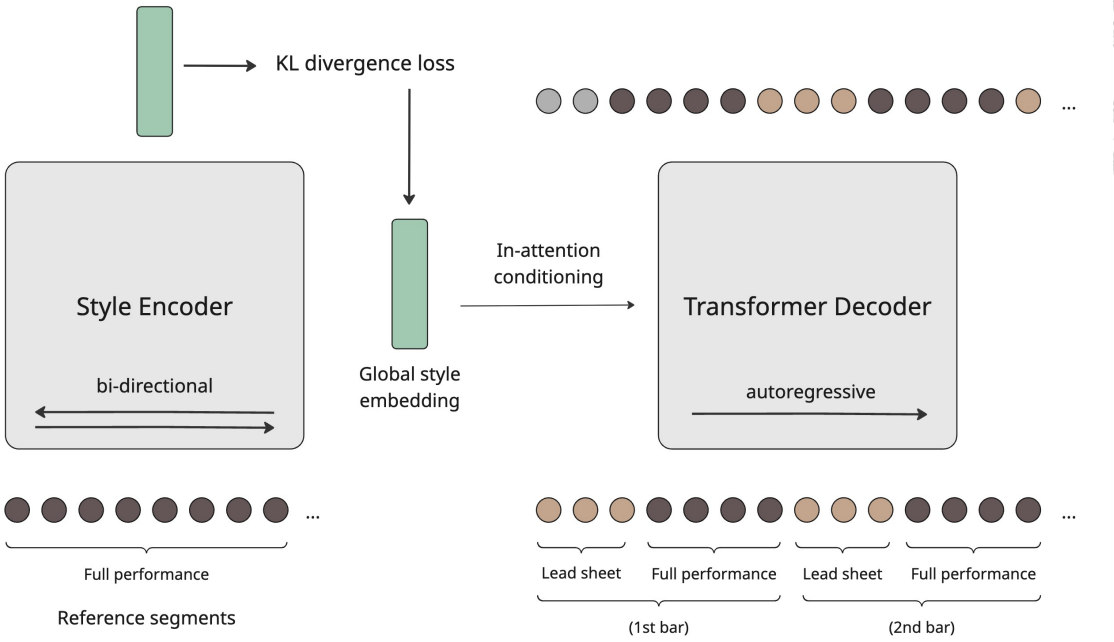


Figure 3.3: Model 3: Encoder-Decoder Transformer with VAE (content: lead sheet tokens, style: reference's full performance embedding)

3.3 Data Representation and Tokenization

3.3.1 Data Source and Conversion

The dataset comprises piano cover performances from two YouTube channels, selected for consistent quality. Audio recordings are converted to MIDI format for symbolic processing, enabling lead sheet extraction and tokenization.

3.3.2 Lead Sheet Extraction

Lead sheets capture the main melody and chord progression from MIDI files. The melody is extracted using the Skyline algorithm, selecting the highest-pitched note per time step, quantized to 8 positions per bar (sub-beats 0, 2, ..., 14 in a 16-sub-beat bar). Chords are identified using the Chorder tool, providing bar-level harmonic annotations

doi:10.6342/NTU202501267

(e.g., C major, G minor).

3.3.3 Symbolic Representation and Tokenization

Music is represented using the REMI framework [22]. For full performances (X_perf, denoted [Track_Midi]), tokens include:

- Bar and Beat tokens (16th-note resolution).
- Note_Pitch (MIDI note number from 21 to 108), Note_Duration (multiples of 16th note), and Note_Velocity (from 40 to 114 in steps of 2).
- Tempo (from 32 to 224 BPM in steps of 3) and Chord tokens (root + quality, 132 types).

For lead sheets (M_lead, denoted [Track_Skyline]), the representation omits Note_Velocity, uses a single global Tempo token (mean of all the Tempo value in full performance), and aligns melody to an 8-step resolution. The vocabulary, including special tokens ([EOS], [Arranger_A], [Arranger_B]), totals 363 tokens, mapped to integer IDs. Sequences are padded or truncated to 1024 tokens. Table 3.1A.1 lists the vocabulary, and Figure 3.4 shows the REMI structure.

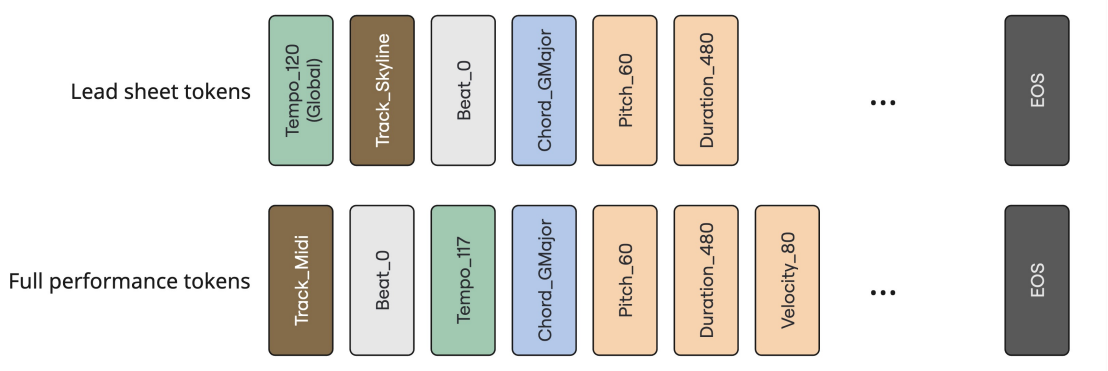


Figure 3.4: REMI sequence for lead sheet & full performance.

doi:10.6342/NTU202501267

3.3.4 Style Representation

Models 1 and 2 use explicit style tokens (e.g., [Arranger_1], [Arranger_2]) within the vocabulary. Model 3 generates a global style embedding from a reference performance's REMI tokens via the style encoder, integrated through in-attention.

3.3.5 Sequence Segmentation

Training samples start at a randomly selected bar, with sequences formatted to 1024 tokens. Model 1 interleaves lead sheet and performance tokens, ensuring content proximity (Figure 3.1). Model 2 and Model 3 process 8-bar lead sheet and performance segments (Figure 3.2, Figure 3.3), with padding or truncation as needed.

3.4 Training Objectives

The training objectives are tailored to each model to optimize performance and ensure content-style disentanglement. For Model 1 and Model 2, the objective is to minimize the negative log-likelihood (NLL) loss for autoregressive token prediction:

$$\mathcal{L}_{\text{NLL}} = -\sum_{t=1}^{T} \log p(y_t | y_{<t}, C)$$

where y_t is the target token and C includes lead sheet and style tokens. Model 3 uses a $\beta - VAE$ [15] objective to enhance disentanglement:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Reconstruction}} + \beta \cdot \mathcal{L}_{\text{KL}}$$

doi:10.6342/NTU202501267

- Reconstruction Loss: An NLL loss for generating performance tokens conditioned on the lead sheet and style embedding: $\mathcal{L}_{\text{Reconstruction}} = -\sum_{t=1}^{T_X} \log p(x_t | x_{<t}, M, z)$
- KL Divergence Loss: Regularizes the style latent space to a standard normal prior, using a "free bits" technique ($\lambda = 0.3$) to prevent posterior collapse and ensure robust disentanglement: $\mathcal{L}_{\text{KL}} = D_{\text{KL}}(q_\phi(z|X_{\text{ref}})||p(z))$. The VAE bottleneck is critical for Model 3, enabling clear separation of content (lead sheet) and accompaniment style.

3.5 Implementation Details

This section provides the specific details regarding model configurations and the training procedures employed in this study, ensuring the reproducibility of our experiments.

3.5.1 Model Configurations and Hyper-parameters

The Transformer architecture forms the backbone of all three proposed models. A consistent set of core hyper-parameters was adopted for all Transformer components across the models (i.e., the decoder in Model 1; the content encoder and decoder in Model 2; and the style encoder and decoder in Model 3). These shared architectural parameters are detailed in Table A.2.

The only model-specific architectural parameter is for Model 3, which is based on a $\beta - VAE$ [15] framework. The dimensionality of its latent style embedding (z_k) is set to 128.

doi:10.6342/NTU202501267

3.5.2 Training Procedure

All models were trained using the Adam optimizer with its standard default parameters ($\beta 1$=0.9, $\beta 2$=0.999, $\epsilon = 10^{-8}$).

A learning rate schedule combining a linear warmup phase with a subsequent cosine annealing decay was implemented. For the initial 200 warmup_steps, the learning rate was linearly increased from a near-zero value to the maximum learning rate of 1.0×10^{-4}. Following the warmup, a CosineAnnealingLR scheduler decayed the learning rate from this maximum value down to a minimum of 5.0×10^{-6} over a cycle of 500,000 steps.

To enhance data diversity and model generalization, an on-the-fly data augmentation strategy was employed. For each training instance, random pitch transposition was applied to the entire musical piece, including both melody and chord information, with the transposition interval uniformly sampled from -6 to +6 semitones.

The batch size was set to 4 for Model 1 and 8 for Model 2 and Model 3. Training for each model was conducted for approximately 1000 epochs, with the final checkpoints for evaluation selected based on the best performance on the validation set. To stabilize the training process, gradient clipping was applied, where the L2 norm of the gradients was clipped to a maximum value of 0.5.

3.5.3 VAE-Specific Training Details (Model 3)

The training of the $\beta - VAE$-based Model 3 incorporated specific regularization strategies. Cyclical KL annealing was applied to the weight (β) of the KL divergence term in the loss function. Initially, β was set to 0 for the first 5,000 training steps to

doi:10.6342/NTU202501267

allow the model to focus on reconstruction. Subsequently, β was linearly increased from 0 to a maximum of 1.0 over repeating 5,000-step cycles. Additionally, the "free bits" [24] technique was used with the hyperparameter λ set to 0.3 to encourage more effective utilization of the latent space.

3.5.4 Software and Hardware

All experiments were conducted using Python 3.8 and the PyTorch framework. Model training was performed on a single NVIDIA GeForce RTX 3090 GPU, with the training process for each of the three models taking approximately one to two days to complete.

doi:10.6342/NTU202501267

Chapter 4 Experiment

This chapter details the dataset specifically compiled and utilized for training, validation and testing the proposed piano accompaniment style transfer models, along with evaluation metrics, experimental setup, and results. The evaluation focuses on style matching (how accurately the generated accompaniments reflect the target arranger's style) and melodic fidelity (how well the core melodic content from the input lead sheet is preserved).

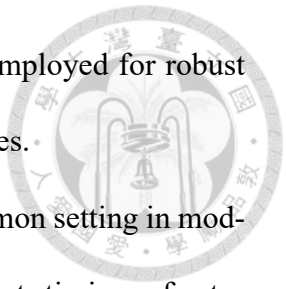
4.1 Dataset

4.1.1 Dataset Preparation

The foundation of our dataset consists of audio recordings of piano cover performances from the Pop2Piano dataset [6], sourced from YouTube channels of two distinct arrangers, referred to as Arranger A and Arranger B, selected for their distinguishable stylistic approaches to popular song accompaniment. The initial processing pipeline, inspired by the Compound Word Transformer [17], converted audio to symbolic MIDI format:

- Audio Transcription: Audio clips were transcribed into raw MIDI-like note events using ByteDance's GiantMIDI-Piano transcription model [26].

doi:10.6342/NTU202501267



- Beat and Downbeat Tracking: The madmom library [1] was employed for robust beat and downbeat detection within the transcribed performances.
- Temporal Alignment: A resolution of 480 ticks per beat, a common setting in modern Digital Audio Workstations (DAWs), was established. Absolute timings of notes were mapped to these ticks, with tempo inferred from beat intervals. Importantly, at this stage, note timings were not quantized, preserving micro-timing deviations.
- Lead Sheet Extraction: As detailed in Section 3.3.2, lead sheets comprising a main melody line and bar-level chord progressions were extracted. Melody extraction utilized the Skyline algorithm, and chord recognition was performed using the Chorder tool.
- Quantization and Event Creation: Then, all relevant musical attributes (e.g., note durations, note velocities for full performances, BPMs for tempo events) were quantized.
- REMI Conversion: Finally, this processed and quantized musical information was converted into sequences of REMI tokens [22], following the event vocabulary and tokenization procedures described in Sections 3.3.3, with an [EOS] token appended.

4.1.2 Dataset Composition and Splitting

The dataset comprises 811 pieces from Arranger A (average 87 bars) and 581 pieces from Arranger B (average 92 bars), totaling approximately 86 hours. The dataset was split using an 8:1:1 ratio (training:validation:test), performed independently for each arranger's collection to ensure proportional representation. The training set contains 1,113 pieces, while the validation set consists of 139 pieces, and the test set includes 140 pieces.

doi:10.6342/NTU202501267

4.1.3 Dataset Statistics

To provide a deeper understanding of the musical characteristics of our dataset and to highlight potential intrinsic stylistic differences between the two arrangers, a comprehensive statistical analysis was conducted, presented in Table 4.1 & 4.2.

	# Pieces	Total Duration	# Bars / Piece	# Pitches / Bar	# REMI Tokens / Piece
Arranger A	811	49.79 hr	87.01 ± 22.07	11.34 ± 6.73	4081.08 ± 1358.68
Arranger B	581	35.72 hr	92.11 ± 21.79	16.02 ± 7.43	5955.39 ± 1304.34

Table 4.1: Dataset Statistics (Overview).

	Rhythmic Intensity	Polyphony	Pitch Range	Avg. Inter-Onset Interval (IOI)
Arranger A	0.46	3.98	49.10 ± 10.31	0.24 ± 0.084
Arranger B	0.56	5.76	55.95 ± 5.77	0.14 ± 0.04

Table 4.2: Dataset Statistics (Feature-based analysis).

These statistics offer quantitative insights into the dataset's general properties and the distinct musical tendencies of the two arrangers, forming a baseline for evaluating style transfer performance.

4.2 Evaluation Metrics

To quantitatively assess the performance of our proposed style transfer models, we evaluate them based on two primary aspects: style matching and melodic fidelity. All objective metrics are computed on 8-bar musical segments extracted from both generated samples and reference pieces.

doi:10.6342/NTU202501267

4.2.1 Objective Metrics

4.2.1.1 Style Matching

We evaluate style matching by comparing the statistical distributions of several musical features between the generated samples and reference corpora representing each target arranger's style. The following features, as identified by MuseMorphose [37], are considered:

- **Rhythmic Intensity**: This metric quantifies the level of rhythmic activity or density within each bar. It is defined as the percentange of sub-beats with at least one note onset, i.e.:

$$s^{\text{rhym}} = \frac{1}{B}\sum_{b=1}^{B}\mathbf{1}(n_{onset,b}, \geq 1)$$

- **Polyphony**: This refers to the average number of simultaneously sounding notes, indicating textural richness. It is measured at the bar-level, the average number of notes hitting (onset) or holding (not yet released) in a sub-beat, i.e.:

$$s^{\text{poly}} = \frac{1}{B}\sum_{b=1}^{B}(n_{onset,b} + n_{hold,b})$$

- **Pitch Range**: This segment-level metric captures the span between the lowest and highest pitches used, which can be a stylistic indicator.

To compare the distributions of these features, we employ two statistical measures:

- **Kullback-Leibler Divergence (KLD)**: KLD measures the divergence between two probability distributions. A lower KLD value between the feature distribution of generated samples (conditioned on a target style, e.g., Arranger A) and the feature

doi:10.6342/NTU202501267

distribution derived from the corpus of Arranger A's original pieces (serving as the ground truth representation of Arranger A's style distribution) indicates a closer stylistic match. Conversely, a higher KLD value when compared to Arranger B's original pieces would be expected.

- **Overlapping Area (OA)**: OA quantifies the similarity between two probability distributions by calculating the overlapping area of their estimated probability density functions. A higher OA value suggests greater similarity. Similar to KLD, we expect a higher OA when comparing generated samples to their target arranger's characteristic style distribution (e.g., $\text{generation}_{\text{styleA}}$ vs. $\text{real}_{\text{styleA}}$) than when compared to a non-target style distribution (e.g., $\text{generation}_{\text{styleA}}$ vs. $\text{real}_{\text{styleB}}$).

By analyzing these KLD and OA scores across the aforementioned features, we can quantitatively assess the models' ability to capture and reproduce the stylistic characteristics of the target arrangers.

4.2.1.2 Melodic Fidelity

Preserving the core melodic content of the input lead sheet within the generated full piano performance is a primary objective. However, in the context of piano covers, different arrangers may introduce subtle variations to a given melody, such as slight shifts in rhythmic placement or minor alterations in note durations, while still retaining the melody's essential identity. To account for this, our evaluation of melodic fidelity focuses on the presence and correct sequential ordering of the pitches from the input lead sheet's melody within the corresponding bars of the generated full performance, rather than requiring exact note-to-note temporal alignment or contiguity.

doi:10.6342/NTU202501267

We employ the Longest Common Subsequence (LCS) algorithm to quantify this aspect of content preservation. The LCS algorithm is applied on a bar-by-bar basis. For each bar, we compare two sequences of pitches:

- The sequence of pitches constituting the melody in the input lead sheet for that bar.
- The sequence of all pitches present in the generated full performance for the corresponding bar.

The LCS algorithm identifies the longest subsequence of pitches that appears in both sequences in the same order, though not necessarily contiguously in the generated performance (as the performance will contain additional accompaniment notes). The melodic fidelity for a given bar is then calculated as:

$$\text{Melodic Fidelity}_{\text{bar}} = \frac{\text{Length}(\text{LCS}(\text{LeadSheetPitches}_{\text{bar}}, \text{GeneratedPitches}_{\text{bar}}))}{\text{Number of Pitches in LeadSheetPitches}_{\text{bar}}} \quad (4.1)$$

This ratio, ranging from 0 to 1, indicates the proportion of lead sheet melody pitches that are present in the correct order within the generated bar. An overall melodic fidelity score for a generated piece (or an 8-bar segment) can be obtained by averaging these bar-level fidelity scores.

4.2.2 Subjective Evaluation

To complement the objective metrics, a subjective listening study was designed and conducted to assess the perceptual quality of the style transfer generated by our models from human listeners' perspectives. This study evaluated aspects such as melodic fidelity,

doi:10.6342/NTU202501267

style matching, and overall musical quality of the generated piano cover arrangements.

Participants: Approximately 38 participants were recruited, encompassing a diverse range of musical backgrounds, categorized into five groups: (1) No prior music knowledge; (2) Less than 1 year of musical experience (self-taught or formal); (3) 1–3 years of musical experience; (4) More than 3 years of musical experience; (5) Currently working in a music-related profession. The listening task was designed to be completed in approximately 20 minutes.

Musical Excerpts and Experimental Design: The study used eight popular songs, not included in the training dataset, for which piano cover arrangements by both Arranger A and Arranger B were available. Two questionnaires were utilized, each containing four songs (two for A → B style transfer, two for B → A). All excerpts were 8 bars. The procedure for the listening study was:

- **Style Familiarization**: Participants listened to representative paired song segments by Arranger A and B to familiarize themselves with their styles.
- **Presentation of Source Piano Cover**: An 8-bar excerpt of the source arranger's performance (e.g., Arranger A) was presented.
- **Evaluation of Transferred Piano Covers**: Participants evaluated multiple 8-bar performances, assessing variations in model-generated outputs and ground truth references.

Participants evaluated performances generated by two models, both conditioned on style tokens: Model 1 (decoder-only) and Model 2 (encoder-decoder with lead sheet embedding), along with the ground truth target performed by the target arranger. Model 3, which relied on an input reference segment to represent style latent, was excluded from the user

doi:10.6342/NTU202501267

study, as the reference segment might not reliably capture the intended target arranger's stylistic characteristics. To mitigate bias, the presentation order of performances was randomized.

Evaluation Questions and Scale: Participants rated each performance on a 5-point Likert scale (1 = strong disagreement/poor, 5 = strong agreement/excellent):

- ***Melodic Fidelity***: To what extent does this performance preserve the melody of the source piano cover?
- ***Style Matching***: To what extent does this performance align with the musical style of the target arranger's piano covers?
- ***Overall Quality***: How would you rate the overall quality of this musical performance?

4.3 Baseline Models

The primary experimental focus of this thesis is a comparative analysis of the three proposed model variants (Model 1, Model 2, and Model 3), each designed to explore different strategies for content representation and style conditioning in piano accompaniment style transfer.

Direct comparisons with existing external baseline models from prior literature were deemed challenging. As discussed in Chapter 2, the field of music style transfer encompasses a wide variety of tasks and musical domains. For instance, Pop2Piano [6] primarily addresses transcription from audio with style control, differing from our MIDI-to-MIDI style transfer objective. Similarly, while "Encoding Musical Style with Transformer Autoencoders" [7] presents highly relevant concepts in learning style embeddings from

doi:10.6342/NTU202501267

symbolic data to control generation, it could not be adapted as a direct baseline due to the lack of a complete open-source implementation for our specific task. Other works focusing on different genre transfers (e.g., classical to jazz) or multi-track instrumentation further complicate direct and fair comparisons. Given these variations in task definition, input modality, and the general lack of adaptable open-source implementations, rigorous head-to-head comparisons were considered impractical for this study.

Therefore, our evaluation centers on the internal comparison of the three proposed models. These models were intentionally designed to encapsulate and investigate different representation strategies (e.g., token-based vs. embedding-based) observed in the literature. In this sense, they effectively serve as controlled baselines for one another, allowing us to draw clear conclusions about the efficacy of these specific design choices for our task.

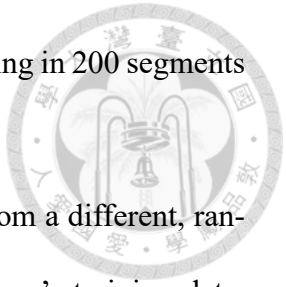
4.4 Experimental Setup

This section details the procedures for generating musical outputs from our trained models and preparing reference data, which are used for the objective and subjective evaluations described in Section 4.2. All generated and reference musical segments are standardized to 8 bars.

To create a corpus of generated samples for evaluation, we selected 100 distinct musical pieces (lead sheets) from the test set. For each piece, 8-bar accompaniments were generated targeting the styles of both Arranger A and Arranger B. The conditioning method varied by model:

- Model 1 and Model 2: Conditioned using style tokens, two distinct 8-bar segments

doi:10.6342/NTU202501267



were generated for each target style per source lead sheet, resulting in 200 segments per style for each model (100 pieces × 2 samples).

- Model 3: Conditioned using style embeddings, each derived from a different, randomly selected 8-bar reference segment from the target arranger's training data, resulting in 200 segments per style. (100 pieces × 2 references).

During generation, nucleus sampling top-p, ($p = 0.9$), temperature ($\tau = 1.2$) decoded output tokens [16]. For Model 3, style embedding (z_k) was set to the mean of the learned posterior distribution ($z_k = \mu_k$), excluding variance for improved quality.

doi:10.6342/NTU202501267

Chapter 5 Results and Discussion

5.1 Overview

Following the evaluation metrics and sample generation in Sections 4.2 and 4.4, this section evaluates our arranger-specific style transfer framework for pop piano covers, comparing three models: Model 1 (token-based, using lead sheets and style tokens), Model 2 (embedding-based, with lead sheet embeddings and style tokens), and Model 3 (embedding-based, with full performance embeddings). Using 200 8-bar segments per style (Arranger A, B) from test set, we assess style matching (rhythmic intensity, polyphony, pitch range) and melodic fidelity objectively, with subjective listener ratings for Model 1 and Model 2. Ablation studies explore Model 1's style token placement and chord usage, and Model 3's reference segments. Similar objective metrics validate the robustness of lead sheets, while subjective results and ablations favor Model 1's simplicity for music production applications.

5.2 Objective Results: Style Matching

The objective results for style matching, were derived by comparing 200 generated 8-bar segments per style against robust style distributions. To ensure statistical stability,

doi:10.6342/NTU202501267

these reference distributions were established by averaging results over five independent random samples of 200 8-bar segments from each arranger's training data. The following analysis focuses on verifying whether the models' generated outputs are statistically closer to their intended target style distribution than to the non-target distribution across several key musical features. A feature-by-feature analysis reveals the models' effectiveness in style discrimination:

- ***Polyphony***: This feature proved to be a key stylistic differentiator that all models successfully captured. As shown by the Overlapping Area (OA) scores in Table 5.1, all models generated outputs that were significantly more similar to their target style distribution than to the non-target one. The difference was clear, with the OA for matched-style comparisons being substantially higher (e.g., by 20%-30%) than for mismatched-style comparisons. Model 1 showed a slight advantage in this regard. A notable observation, however, appeared in the Kullback-Leibler Divergence (KLD) scores for an ablation version of Model 1 trained without chord information, as shown in Table 5.4. For this "chord-less" model, the KLD scores showed an unusual bias: its generated outputs, regardless of conditioning, consistently had a higher KLD when compared to Arranger B's data than to Arranger A's. This may point to a limitation of the KLD metric in this specific case, where it might be sensitive to underlying data artifacts when harmonic guidance is absent.
- ***Rhythmic Intensity***: All models also performed well on this feature. The results from both OA and KLD (Table 5.2 and Table 5.5) consistently show that the generated music is statistically closer to the intended target style. While the distinction between the target and non-target styles is more subtle here compared to polyphony, this is likely because the two arrangers' styles are inherently more similar for this

doi:10.6342/NTU202501267

particular feature in our dataset.

- ***Pitch Range***: The models also successfully captured the distinct pitch range characteristics of each arranger, as clearly demonstrated by the OA scores. As shown in Table 5.3, all models' outputs align well with their intended target style distribution. Interestingly, the KLD metric (Table 5.6) did not consistently reflect this trend, showing a bias towards Arranger A's distribution in most cases. While this discrepancy points to potential limitations or different sensitivities of the KLD metric for this specific feature, the strong, positive results from the OA metric confirm that the models did effectively learn to replicate the target pitch range styles.

Model	**Gen A vs. Train A**	**Gen A vs. Train B**	**Gen B vs. Train A**	**Gen B vs. Train B**
Model 1	0.92 ± 0.01	0.58 ± 0.01	0.57 ± 0.02	0.95 ± 0.01
bar-level style	0.94 ± 0.01	0.6 ± 0.01	0.64 ± 0.02	0.92 ± 0.01
w/o chord	0.89 ± 0.01	0.56 ± 0.01	0.61 ± 0.02	0.95 ± 0.01
Model 2	0.95 ± 0.01	0.6 ± 0.01	0.65 ± 0.02	0.93 ± 0.01
Model 3	0.94 ± 0.01	0.64 ± 0.01	0.72 ± 0.02	0.88 ± 0.01
adjacent	0.89 ± 0.01	0.71 ± 0.01	0.61 ± 0.02	0.96 ± 0.01

Table 5.1: Objective evaluation results: Overlapping Area - Polyphony.

Model	**Gen A vs. Train A**	**Gen A vs. Train B**	**Gen B vs. Train A**	**Gen B vs. Train B**
Model 1	0.91 ± 0.01	0.75 ± 0.03	0.8 ± 0.02	0.91 ± 0.02
bar-level style	0.86 ± 0.01	0.72 ± 0.03	0.82 ± 0.02	0.9 ± 0.02
w/o chord	0.88 ± 0.01	0.73 ± 0.03	0.81 ± 0.02	0.91 ± 0.03
Model 2	0.88 ± 0.01	0.81 ± 0.03	0.82 ± 0.02	0.92 ± 0.02
Model 3	0.86 ± 0.01	0.72 ± 0.03	0.83 ± 0.02	0.85 ± 0.03
adjacent	0.86 ± 0.02	0.79 ± 0.03	0.73 ± 0.02	0.91 ± 0.02

Table 5.2: Objective evaluation results: Overlapping Area - Rhythmic intensity.

Model	**Gen A vs. Train A**	**Gen A vs. Train B**	**Gen B vs. Train A**	**Gen B vs. Train B**
Model 1	0.93 ± 0.02	0.57 ± 0.01	0.56 ± 0.03	0.96 ± 0.01
bar-level style	0.95 ± 0.01	0.59 ± 0.01	0.62 ± 0.03	0.92 ± 0.01
w/o chord	0.91 ± 0.02	0.55 ± 0.01	0.6 ± 0.03	0.95 ± 0.01
Model 2	0.95 ± 0.01	0.59 ± 0.01	0.63 ± 0.02	0.93 ± 0.0
Model 3	0.93 ± 0.02	0.63 ± 0.01	0.7 ± 0.03	0.87 ± 0.01
adjacent	0.88 ± 0.02	0.7 ± 0.01	0.6 ± 0.03	0.96 ± 0.01

Table 5.3: Objective evaluation results: Overlapping Area - Pitch range.

doi:10.6342/NTU202501267

Model	Gen A vs. Train A	Gen A vs. Train B	Gen B vs. Train A	Gen B vs. Train B
Model 1	0.03 ± 0.02	0.38 ± 0.23	0.72 ± 0.22	0.11 ± 0.1
bar-level style	0.04 ± 0.04	0.62 ± 0.29	0.31 ± 0.13	0.1 ± 0.13
w/o chord	1.39 ± 0.14	2.62 ± 0.27	0.1 ± 0.05	0.62 ± 0.33
Model 2	0.13 ± 0.09	0.17 ± 0.14	0.75 ± 0.22	0.13 ± 0.11
Model 3	0.11 ± 0.07	0.18 ± 0.16	0.6 ± 0.2	0.1 ± 0.07
adjacent	0.25 ± 0.11	0.11 ± 0.09	0.68 ± 0.21	0.09 ± 0.08

Table 5.4: Objective evaluation results: Kullback-Leibler Divergence - Polyphony.

Model	Gen A vs. Train A	Gen A vs. Train B	Gen B vs. Train A	Gen B vs. Train B
Model 1	0.04 ± 0.01	0.21 ± 0.03	0.1 ± 0.03	0.02 ± 0.01
bar-level style	0.06 ± 0.01	0.27 ± 0.04	0.08 ± 0.03	0.03 ± 0.01
w/o chord	0.05 ± 0.01	0.23 ± 0.04	0.08 ± 0.03	0.03 ± 0.01
Model 2	0.04 ± 0.01	0.12 ± 0.03	0.1 ± 0.04	0.03 ± 0.01
Model 3	0.06 ± 0.01	0.19 ± 0.04	0.06 ± 0.02	0.06 ± 0.02
adjacent	0.06 ± 0.02	0.11 ± 0.03	0.18 ± 0.05	0.01 ± 0.0

Table 5.5: Objective evaluation results: Kullback-Leibler Divergence - Rhythmic intensity.

Model	Gen A vs. Train A	Gen A vs. Train B	Gen B vs. Train A	Gen B vs. Train B
Model 1	0.07 ± 0.0	0.19 ± 0.0	0.33 ± 0.0	1.02 ± 0.0
bar-level style	0.15 ± 0.0	0.24 ± 0.0	0.39 ± 0.0	1.05 ± 0.0
w/o chord	0.28 ± 0.0	0.89 ± 0.0	0.68 ± 0.0	1.46 ± 0.0
Model 2	0.19 ± 0.0	0.19 ± 0.0	0.41 ± 0.0	1.04 ± 0.0
Model 3	0.17 ± 0.0	0.63 ± 0.0	0.39 ± 0.0	1.11 ± 0.0
adjacent	0.23 ± 0.0	0.11 ± 0.0	0.18 ± 0.0	0.56 ± 0.0

Table 5.6: Objective evaluation results: Kullback-Leibler Divergence - Pitch range.

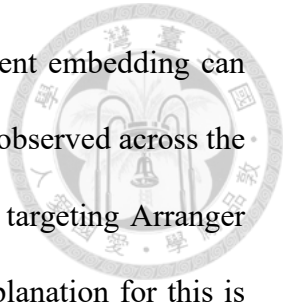
Model	Arranger A	Arranger B
Model 1	0.97 ± 0.1	0.91 ± 0.17
bar-level style	0.97 ± 0.1	0.96 ± 0.1
w/o chord	0.95 ± 0.13	0.94 ± 0.13
Model 2	0.91 ± 0.23	0.89 ± 0.23
Model 3	0.92 ± 0.16	0.88 ± 0.2
adjacent	0.93 ± 0.15	0.94 ± 0.15

Table 5.7: Objective evaluation results: Average melodic fidelity.

5.3 Objective Results: Melodic Fidelity

The aggregated results, shown in Table 5.7, indicate that all models achieved high melodic fidelity in preserving the content from the input lead sheets. Model 1 (token-based) led with an overall average fidelity of 91%-97%, followed closely by Model 3 (88%-94%) and Model 2 (89%-91%). The slightly lower performance of Model 2 may

doi:10.6342/NTU202501267



suggest that encoding the entire lead sheet into a single global content embedding can result in minor information loss. Furthermore, a consistent trend was observed across the models: melodic fidelity scores were generally slightly lower when targeting Arranger B's style compared to when targeting Arranger A's. A plausible explanation for this is that Arranger B's own performance style tends to incorporate more melodic variations and embellishments. It is likely that our models, in learning to emulate this style, also learned to introduce similar melodic deviations not present in the original, strict lead sheet, thus resulting in a marginally lower LCS ratio. This suggests the models not only transferred the accompaniment style but also subtle aspects of the melodic performance style.

5.4 Subjective Results

Subjective evaluations, conducted with 38 listeners rating Model 1, Model 2, and ground truth performances on a 5-point Likert scale, revealed clear preferences, with detailed scores presented in Table 5.8 and Table 5.9. Focusing first on Style Matching, the token-based Model 1 (3.24/5) was rated as significantly more effective at capturing the target arranger's style than Model 2 (2.63/5). Beyond just matching style, Model 1 also excelled at content preservation; for Melodic Fidelity, it again outperformed Model 2 (3.35/5 vs. 2.75/5) and achieved a rating comparable to that of the ground truth human performances (3.53/5), indicating robust melody preservation. This strong performance was reflected in the Overall Quality ratings, where Model 1 (3.28/5) was again clearly preferred over Model 2 (2.91/5) and its score approached the high standard set by the ground truth (3.27/5). The consistent underperformance of Model 2 across all subjective criteria suggests that its content-embedding approach may lead to outputs that are perceived as less distinctive and faithful. An interesting phenomenon was observed in this criterion

doi:10.6342/NTU202501267

when targeting Arranger B's style: listeners rated Model 1's melodic preservation (3.5/5) as slightly higher than that of the actual ground truth performance by Arranger B (3.34/5). This may be because the real performance by Arranger B incorporates more stylistic variations and embellishments on the core melody; some listeners might have perceived these artistic choices as a deviation, whereas Model 1's stricter adherence to the input lead sheet was perceived as more "faithful".

Model	Melodic Fidelity	Style Matching	Overall Quality
Model 1	3.35 ± 0.95	**3.24** ± 1.01	**3.28** ± 0.97
Model 2	2.75 ± 1.26	2.63 ± 1.23	2.91 ± 1.14
Real Data	**3.53** ± 1.04	3.2 ± 1.16	3.27 ± 1.07

Table 5.8: User study MOS results. (Aggregated overall.)

Model	Melodic Fidelity		Style Matching		Overall Quality	
	Arranger A	Arranger B	Arranger A	Arranger B	Arranger A	Arranger B
Model 1	3.2 ± 0.95	**3.5** ± 0.93	3.21 ± 0.94	**3.26** ± 1.08	3.28 ± 0.87	**3.28** ± 1.07
Model 2	2.89 ± 1.26	2.61 ± 1.24	2.78 ± 1.2	2.49 ± 1.25	2.95 ± 1.06	2.88 ± 1.23
Real Data	**3.72** ± 0.87	3.34 ± 1.15	**3.32** ± 1.12	3.09 ± 1.2	**3.47** ± 1.01	3.07 ± 1.1

Table 5.9: User study MOS results. (Presented by target arrangers.)

5.5 Discussion

The experimental results present a key insight: while all three models achieved comparable performance on objective style-matching metrics—validating our lead sheet-based framework for effective content disentanglement—a clear preference for the simplest, token-based architecture (Model 1) emerged in human evaluation. Its subjective superiority in style matching, melodic fidelity, and overall quality, combined with its computational efficiency (faster training, fewer parameters), suggests that for this specific task, a direct token-based conditioning approach is more effective than compressing content into a fixed embedding, which may risk information loss as seen in Model 2. Furthermore, our analysis revealed limitations in the objective metrics themselves, such as a consistent pitch

doi:10.6342/NTU202501267

range bias across all models. This underscores the critical role of subjective evaluation in capturing the subtle nuances of musical style that current metrics may miss. While Model 1 proved to be the most practical solution in this study, the robustness of Model 3's style encoder indicates that latent space modeling remains a promising, albeit more complex, paradigm for future exploration.

doi:10.6342/NTU202501267

doi:10.6342/NTU202501267

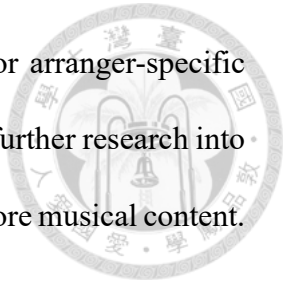
Chapter 6 Conclusion

This thesis investigated the challenge of arranger-specific style transfer for single-track piano accompaniments. We proposed and systematically evaluated a framework grounded on the use of lead sheets as a robust content anchor, comparing three distinct Transformer-based architectures to assess the efficacy of token-based versus embedding-based strategies for representing musical content and style.

Our primary finding is that a straightforward token-based model, built upon a strong lead sheet content representation, proved to be a highly effective and computationally efficient solution for this task. While objective metrics indicated comparable performance across all models, validating the overall framework, the token-based approach was clearly preferred by human listeners, scoring significantly higher in style matching, melodic fidelity, and overall quality. This work's main contribution is providing empirical evidence that for transferring between a known set of nuanced styles, an explicit content representation can be more critical than complex latent space modeling.

The findings from this research open up several promising avenues for future work, primarily centered on representation learning for content-style disentanglement. A key direction is to further explore how to best represent content and style for different transfer tasks. This involves designing representations that are tailored to the specific musical

doi:10.6342/NTU202501267



characteristics of the target domain, whether it be genre, emotion, or arranger-specific patterns. For example, the success of the lead sheet anchor motivates further research into other structured, symbolic representations that can effectively model core musical content.

Furthermore, while our simpler token-based method excelled in this study, the disentangling capabilities of Transformer-VAEs (as explored in Model 3) remain a critical area for development, especially for enabling more generalized, zero-shot style transfer. Future research could investigate more advanced techniques to achieve a cleaner separation of style and content factors within the latent space. This includes exploring more sophisticated reference segment designs for style encoders or integrating adversarial objectives to improve the disentanglement process. In summary, this thesis not only provides a practical solution for arranger-specific style transfer but also contributes to the broader understanding of how representation choices impact controllable music generation.

doi:10.6342/NTU202501267

References

[1] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer. Madmom: A new python audio and music signal processing library. In Proceedings of the 24th ACM international conference on Multimedia, pages 1174–1178, 2016.

[2] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. Modeling temporal dependencies in high-dimensional sequences: Application to polyphonic music generation and transcription. arXiv preprint arXiv:1206.6392, 2012.

[3] J.-P. Briot, G. Hadjeres, and F.-D. Pachet. Deep learning techniques for music generation–a survey. arXiv preprint arXiv:1709.01620, 2017.

[4] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer. Midi-vae: Modeling dynamics and instrumentation of music with applications to style transfer. arXiv preprint arXiv:1809.07600, 2018.

[5] G. Brunner, Y. Wang, R. Wattenhofer, and S. Zhao. Symbolic music genre transfer with cyclegan. In 2018 ieee 30th international conference on tools with artificial intelligence (ictai), pages 786–793. IEEE, 2018.

[6] J. Choi and K. Lee. Pop2piano : Pop audio-based piano cover generation, 2023.

[7] K. Choi, C. Hawthorne, I. Simon, M. Dinculescu, and J. Engel. Encoding musi-

doi:10.6342/NTU202501267

cal style with transformer autoencoders. In International conference on machine learning, pages 1899–1908. PMLR, 2020.

[8] O. Cífka, U. Şimşekli, and G. Richard. Supervised symbolic music style translation using synthetic data. arXiv preprint arXiv:1907.02265, 2019.

[9] O. Cífka, U. Şimşekli, and G. Richard. Groove2groove: One-shot music style transfer with supervision from synthetic data. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 28:2638–2650, 2020.

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

[11] H.-W. Dong, W.-Y. Hsiao, L.-C. Yang, and Y.-H. Yang. Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In Proceedings of the AAAI conference on artificial intelligence, volume 32, 2018.

[12] D. Eck and J. Schmidhuber. A first look at music composition using lstm recurrent neural networks. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 103(4):48–56, 2002.

[13] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. Advances in neural information processing systems, 27, 2014.

[14] G. Hadjeres, F. Pachet, and F. Nielsen. Deepbach: a steerable model for bach chorales generation. In International conference on machine learning, pages 1362–1371. PMLR, 2017.

doi:10.6342/NTU202501267

[15] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner. beta-vae: Learning basic visual concepts with a constrained variational framework. In International conference on learning representations, 2017.

[16] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi. The curious case of neural text degeneration. arXiv preprint arXiv:1904.09751, 2019.

[17] W.-Y. Hsiao, J.-Y. Liu, Y.-C. Yeh, and Y.-H. Yang. Compound word transformer: Learning to compose full-song music over dynamic directed hypergraphs. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 178–186, 2021.

[18] W.-N. Hsu, Y. Zhang, and J. Glass. Learning latent representations for speech generation and transformation. arXiv preprint arXiv:1704.04222, 2017.

[19] Z. Hu, Z. Yang, R. R. Salakhutdinov, L. Qin, X. Liang, H. Dong, and E. P. Xing. Deep generative models with learnable knowledge constraints. Advances in Neural Information Processing Systems, 31, 2018.

[20] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, N. Shazeer, C. Hawthorne, A. M. Dai, M. D. Hoffman, and D. Eck. Music transformer: Generating music with long-term structure. arXiv preprint arXiv:1809.04281, 2018.

[21] J. Huang, K. Chen, and Y.-H. Yang. Emotion-driven piano music generation via two-stage disentanglement and functional representation. arXiv preprint arXiv:2407.20955, 2024.

[22] Y.-S. Huang and Y.-H. Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In Proceedings of the 28th ACM international conference on multimedia, pages 1180–1188, 2020.

doi:10.6342/NTU202501267

[23] Y.-N. Hung, I. Chiang, Y.-A. Chen, Y.-H. Yang, et al. Musical composition style transfer via disentangled timbre representations. arXiv preprint arXiv:1905.13567, 2019.

[24] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. Advances in neural information processing systems, 29, 2016.

[25] D. P. Kingma, M. Welling, et al. Auto-encoding variational bayes, 2013.

[26] Q. Kong, B. Li, X. Song, Y. Wan, and Y. Wang. High-resolution piano transcription with pedals by regressing onset and offset times. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 29:3707–3717, 2021.

[27] W. T. Lu, L. Su, et al. Transferring the style of homophonic music using recurrent neural networks and autoregressive model. In ISMIR, pages 740–746, 2018.

[28] H. H. Mao, T. Shin, and G. Cottrell. Deepj: Style-specific music generation. In 2018 IEEE 12th international conference on semantic computing (ICSC), pages 377–382. IEEE, 2018.

[29] N. Mor, L. Wolf, A. Polyak, and Y. Taigman. A universal music translation network. arXiv preprint arXiv:1805.07848, 2018.

[30] A. Pati and A. Lerch. Is disentanglement enough? on latent representations for controllable music generation. arXiv preprint arXiv:2108.01450, 2021.

[31] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck. A hierarchical latent vector model for learning long-term structure in music. In International conference on machine learning, pages 4364–4373. PMLR, 2018.

doi:10.6342/NTU202501267

[32] H. H. Tan and D. Herremans. Music fadernets: Controllable music generation based on high-level features via low-level feature modelling. arXiv preprint arXiv:2007.15474, 2020.

[33] P. M. Todd. A connectionist approach to algorithmic composition. Computer Music Journal, 13(4):27–43, 1989.

[34] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, K. Kavukcuoglu, et al. Wavenet: A generative model for raw audio. arXiv preprint arXiv:1609.03499, 12, 2016.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. Advances in neural information processing systems, 30, 2017.

[36] S.-L. Wu and Y.-H. Yang. Compose & embellish: Well-structured piano performance generation via a two-stage approach. In ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[37] S.-L. Wu and Y.-H. Yang. Musemorphose: Full-song and fine-grained piano music style transfer with one transformer vae. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31:1953–1967, 2023.

[38] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang. Midinet: A convolutional generative adversarial network for symbolic-domain music generation. arXiv preprint arXiv:1703.10847, 2017.

[39] H. Zhang and S. Dixon. Disentangling the horowitz factor: Learning content and style from expressive piano performance. In ICASSP 2023-2023 IEEE International

doi:10.6342/NTU202501267

Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 1–5. IEEE, 2023.

[40] J. Zhao and G. Xia. Accomontage: Accompaniment arrangement via phrase selection and style transfer. arXiv preprint arXiv:2108.11213, 2021.

doi:10.6342/NTU202501267

Appendix A — Experiment

A.1 REMI Vocabulary

Event Type	Description	# Tokens
Bar	Marks the beginning of a new bar.	1
Beat	Discrete beat/sub-beat position within a 4/4 bar, in 16th note resolution. For lead sheet events, quantized to 8th note resolution.	16
Tempo	Represents quantized BPM values, e.g., from 32 bpm to 224 bpm in steps of 3 bpm.	65
Note_Pitch	MIDI note numbers (pitch) from 21 to 108	88
Note_Duration	multiples (1-16 times) of 16th note	17
Note_Velocity	MIDI velocity (loudness), from 40 to 114 in steps of 2	38
Chord	chord markings (root & quality) e.g., G minor seventh	133
EOS	End-of-sequence or end-of-segment token.	1
Track_Skyline	Identifier for lead sheet data segments.	1
Track_Midi	Identifier for full performance data segments.	1
Arranger_A	Style token representing Arranger A.	1
Arranger_B	Style token representing Arranger B.	1

Table A.1: The REMI vocabulary used to represent piano cover songs in our dataset.

doi:10.6342/NTU202501267

A.2 Model Configurations

Parameter	Value	Notes
Common Transformer Architecture		
# Self-Attention Layers	12	
Token Embedding Dimension	512	
Hidden State Dimension	512	
# Encoder Layers	12	(For Models 2 & 3's encoders)
# Decoder Layers	12	(For all models' decoders)
# Self-Attention Heads (n_{head})	8	(For both encoder & decoder)
Feed-forward Dimension (d_{ff})	2048	(For both encoder & decoder)
Dropout Rate	0.1	
VAE Latent Dimension (z_k)	128	Model 3
Optimizer & Learning Rate		
Optimizer	Adam	
Adam $\beta 1$	0.9	(Default)
Adam $\beta 2$	0.999	(Default)
Adam ϵ	$1\times10-8$	(Default)
Max Learning Rate (max_lr)	$1.0\times10-4$	
Min Learning Rate (min_lr for Cosine)	$5.0\times10-6$	
Warmup Steps	200	(Linear warmup)
Cosine Annealing T_max / lr_decay_steps	500,000 steps	(Steps for one decay cycle)
Training Hyperparameters		
Batch Size (Model 1)	4	
Batch Size (Model 2 & 3)	8	
Gradient Clipping Norm	0.5	(L2 norm)
Total Training Epochs (approx.)	$\tilde{1}000$	(Best checkpoint selected)
VAE-Specific Training (Model 3)		
KL Annealing: no_kl_steps	5,000 steps	(β=0 during these steps)
KL Annealing: kl_cycle_steps	5,000 steps	(For one cycle of β ramp-up)
KL Annealing: kl_max_beta	1	
Free Bits λ	0.3	

Table A.2: Model configurations for all models in our study.

doi:10.6342/NTU202501267