Etude: Piano Cover Generation with a Three-Stage Approach — Extract, strucTUralize, and DEcode

1st Tse-Yang Chen
Dept. of Information Management
National Taiwan University
Taipei, Taiwan
r12725050@ntu.edu.tw

2nd Yuh-Jzer Joung
Dept. of Information Management
National Taiwan University
Taipei, Taiwan
joung@ntu.edu.tw

***Abstract*—Piano cover generation aims to automatically transform a pop song into a piano arrangement. While numerous deep learning approaches have been proposed, existing models often fail to maintain structural consistency with the original song, likely due to the absence of beat-aware mechanisms or the difficulty of modeling complex rhythmic patterns. Rhythmic information is crucial, as it defines structural similarity (e.g., tempo, BPM) and directly impacts the overall quality of the generated music.**

In this paper, we introduce Etude, a three-stage architecture consisting of Extract, strucTUralize, and DEcode stages. By pre-extracting rhythmic information and applying a novel, simplified REMI-based tokenization, our model produces covers that preserve proper song structure, enhance fluency and musical dynamics, and support highly controllable generation through style injection. Subjective evaluations with human listeners show that Etude substantially outperforms prior models, achieving a quality level comparable to that of human composers.

***Index Terms*—automatic piano cover generation, music generation, music information retrieval, automatic music transcription.**

I. Introduction

Piano cover generation—the artistic rearrangement of a song into a piano performance—is a popular form of musical expression that thrives on social media and has grown into a commercially viable field. Yet producing a high-quality piano cover remains challenging, requiring both analysis of the original song's musical features (e.g., harmony and rhythm) and sufficient music theory knowledge to create a coherent, aesthetically pleasing arrangement.

Numerous studies [1]–[5] have explored Automatic Piano Cover Generation (APCG) with deep learning. These methods often emulate the two-step creative process of human composers: first analyzing music-theoretical features such as harmony, melody, and rhythm, and then recombining them to synthesize a new interpretation within the piano's 88-key range. Nevertheless, a substantial quality gap remains between model-generated and human-composed covers, as reflected in subjective evaluation scores, suggesting that the task's complexity and nuance have yet to be fully captured.

A closer examination reveals a key source of this quality gap: prior models struggle to capture a song's rhythmic and structural framework. Their outputs frequently lack consistency in rhythm, tempo, or time signature—with some models

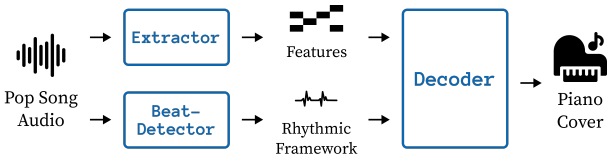

Fig. 1. The architecture of our proposed Etude framework, which comprises three main components: an Extractor for the extract stage, a Beat-Detector for the structuralize stage, and a Decoder for the decode stage.

[1], [4], [5] relying on time-based rather than beat-based frameworks, resulting in off-beat notes and degraded quality. We attribute this shortcoming to an architectural limitation: because beat tracking is notoriously difficult in music information retrieval (MIR), asking models to implicitly learn a song's structure while simultaneously extracting other musical features creates a compounded challenge that undermines their effectiveness.

To address this issue, we introduce **Etude**, a novel architecture that builds upon the two-stage concept of PiCoGen [2], [3], but further modularizes the task by explicitly separating the beat detection process. As illustrated in Figure 1, the name **Etude** reflects the model's three distinct stages: the **E**xtract stage for extracting music-theoretical features, followed by the struc**TU**ralize stage for deriving the rhythmic framework, and concluding with the **DE**code stage where all features are combined to generate the final output. The key advantages of our approach are twofold. First, by explicitly extracting the rhythmic framework, our model guarantees structural consistency with the original song, which significantly enhances overall quality and musicality. Second, our architecture allows for the injection of specified styles, a feature that increases both the controllability of the output and its practical application value.

In summary, by disentangling the learning of structure from the generation of notes, Etude produces structurally sound, stylistically controllable, and high-quality piano covers, demonstrating state-of-the-art performance in extensive evaluations where it substantially outperformed previous methods and achieved a quality closer to human-composed arrangements. All code and audio demonstrations are available on

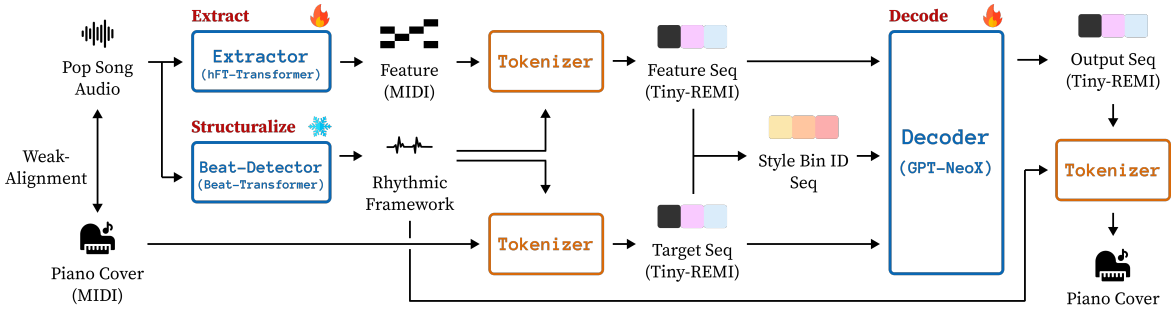

Fig. 2. The overall architecture of **Etude**, where an **Extractor** derives harmonic features, a pre-trained **Beat-Detector** provides the rhythmic framework for tokenization and de-tokenization, and a **Decoder** receives these inputs along with style prompts. The Extractor and Decoder are trained **separately**.

our project page.[1]

II. Background

The task of Automatic Piano Cover Generation (APCG) presents several unique challenges for deep learning models. Among these, a primary requirement is the ability to comprehend long-term musical context spanning multiple measures [6], which hinges on the development of an effective symbolic music representation [7]. Furthermore, to enhance the model's practical utility and user control, incorporating mechanisms from conditional music style transfer is essential [8].

Modern symbolic music generation is fundamentally built on event-based representations. An early and intuitive approach is the piano roll, which encodes music as a 2D pitch–time matrix and is well-suited to image-based models such as CNNs [9]–[11]. However, it lacks explicit note-off information, making it difficult to distinguish between long sustained notes and repeated short notes [7]. Event-based sequences, by contrast, use MIDI-like tokens (e.g., 'Note On' and 'Note Off') that are more explicit but come with limitations: they struggle to represent rhythmic structure clearly, are inefficient to model, and may produce artifacts such as hanging notes. A major advancement came with REMI [12], which introduced *Bar* and *Position* tokens to provide a metrical grid for rhythm. While effective, REMI often generates very long sequences, posing challenges for Transformer-based models. To address this, the Compound Word (CP) representation [13] was proposed, grouping related events into 'words' to shorten sequence length. These beat-aware and compact representations have proven crucial for enabling state-of-the-art models like the Transformer to capture long-range musical dependencies.

Beyond representation, controllable music style transfer has emerged as a central research focus. Early work (e.g., [14]) adapted steerable architectures such as DeepBach [15] for homophonic accompaniment style transfer. A persistent challenge is the scarcity of aligned data, motivating predominantly unsupervised approaches. VAE-based methods, such as MIDI-VAE [8], addressed this by disentangling style in a shared latent space with classifier guidance, while GAN-based frameworks, e.g., CycleGAN [16], leveraged image processing techniques for symbolic genre transfer. More recent research emphasizes attribute-based control, with systems like Muse-Morphose [17], inspired by Music FaderNets [18], enabling fine-grained manipulation of perceptual dimensions such as rhythmic intensity and polyphonic density.

Although these techniques provide a strong foundation, their application to APCG exposes several key challenges. PiCoGen [2], [3] established the two-stage paradigm: extracting core features from a song into a simplified intermediate representation and then generates a piano cover. However, this intermediate step often acts as an information bottleneck, losing much of the original audio's nuance. Data alignment further compounds the problem: strong-alignment methods [1] often induce audio artifacts, whereas weak-alignment approaches [3] remain limited by inaccuracies in beat-tracking. In contrast, models like AMT-APC [4] utilize a more direct transcription approach that captures richer features but often at the cost of explicit high-level structural modeling.

A common limitation across these methodologies is the lack of **structural consistency**. Data artifacts, information bottlenecks, and the inherent difficulty of implicitly modeling rhythm all contribute to failures in preserving the original song's rhythmic framework—tempo, bar, and beat alignment. This gap in structural integrity forms the central motivation for our work.

III. Methodology

Our proposed framework, Etude, is designed to address the key challenges in APCG, particularly in ensuring structural consistency and enabling stylistic control. The architecture, illustrated in Figure 2, modularizes the generation process into three distinct stages: Extract, Structuralize, and Decode.

A. Extract Stage

The goal of the Extract stage is to overcome the information bottleneck found in prior two-stage models [2], [3], which often use simplified lead sheet [19] representations that lose significant musical nuance. To address this, we produce a rich and dense feature representation that captures all potentially salient events from the source audio. Our Extractor's architecture is adapted from the AMT-APC [4], which is based on

[1]https://xiugapurin.github.io/Etude/

the hFT-Transformer [20]. We use the source audio as input but modify the model's loss sampling parameter (θ^{matrix}) to encourage the transcription of a dense map of musical events rather than a sparse, playable arrangement. The final output is a MIDI-like feature sequence that provides a comprehensive and detailed foundation for the Decoder.

B. Structuralize Stage

The Structuralize stage is central to ensuring structural correctness. Running in parallel with the Extractor, it employs a pre-trained Beat-Transformer [21] to analyze the source audio and extract the precise timings of all beats and downbeats. This information forms a definitive rhythmic framework F_{beat}, containing tempo, time signature, and measure boundaries, that serves as an immutable structural ground truth for the entire process. This framework underpins our decoupled approach: it provides all necessary metrical information for the tokenization of symbolic data and the final reconstruction of a structurally coherent MIDI file from the model's output (see Section III-D).

C. Decode Stage

The Decode stage employs a Transformer-based model to synthesize the final piano cover. Its primary goal is to translate the dense feature sequence (X) from the Extractor into the target symbolic piano cover sequence (Y), conditioned on a set of style attributes that guide the arrangement process.

To teach the model this complex translation task, we employ a bar-wise mix [2], [3], [22] training strategy. For each song, the feature sequence X and target sequence Y are segmented by measure and interleaved into a single sequence of the form $[X_1, Y_1, X_2, Y_2, ...]$. To distinguish between the two sources, each token is accompanied by a parallel Class ID (`SRC` for feature tokens, `TGT` for target tokens). The model is trained to predict the target bar Y_i, using its corresponding feature bar X_i and the context of the preceding four mixed bars. This four-bar context window is designed to capture the common phrasing structures found in popular music, thereby enhancing the musical continuity of the output.

To enable controllable generation and address the one-to-many nature of the arrangement task, we introduce **style vectors** that explicitly describe the relationship between a feature bar (X_i) and its corresponding target bar (Y_i). We designed three relative style attributes inspired by prior work [17], [18] on controllable generation:

- **Relative Polyphony**: This measures the change in the average number of notes per unique time event. It reflects a shift in harmonic texture; for example, a high value indicates a transformation towards denser block chords, while a low value suggests a shift towards sparser textures like arpeggios.
- **Relative Rhythmic Intensity**: This measures the change in the density of rhythmic events over time. A higher value results in a more complex and rhythmically active phrase.
- **Relative Note Sustain**: This measures the change in the average note duration, controlling the output's articulation character from legato (high) to staccato (low).

These relative attributes are calculated for every bar pair in the training set and discretized into a small number of bins based on their statistical distribution. During training and inference, each token in a bar-pair sequence is assigned the three corresponding style bin IDs for that bar. These IDs are then passed through separate embedding layers, concatenated, and projected by a linear layer to form a single style embedding vector. This style embedding is added to the token and class ID embeddings to form the final input to the Transformer Decoder. This mechanism effectively conditions the entire generation process on the desired stylistic transformation, allowing for fine-grained, bar-level control over the final output's characteristics.

D. Data Representation

The choice of data representation is critical for the success of a symbolic music generation model. While prior works like Pop2Piano [1] used raw MIDI event sequences, which are difficult for sequence models to learn, subsequent research [2], [3] has gravitated towards more structured representations [12], [13]. However, the standard REMI tokenizer, designed for general-purpose music generation, includes tokens for `Tempo`, `Chord`, and `Velocity`. We argue that for the APCG task, forcing the Decoder to learn these attributes introduces unnecessary complexity. For instance, tempo information is already explicitly handled by our pre-extracted rhythmic framework (F_{beat}), while other attributes like chords and velocity are not essential for learning the core structural translation task that is our focus.

To address this, we designed **Tiny-REMI**, a minimal and efficient token set tailored specifically for the APCG task. Our representation consists of five token classes. `Bar` tokens ([BOS], [EOS]) are special tokens used to mark the beginning and end of each measure. `Pos` is a metric-related token indicating the 16th-note offset within a measure. The remaining tokens are note-related: `Note` for note pitches (88 values), `Duration` for note durations (10 common values based on 16th-note units), and `Grace` for representing grace notes (± 1 semitone). Figure 3 provides an example of a musical measure tokenized in this format. By removing redundant tokens, this minimalist vocabulary simplifies the learning task for the Decoder, allowing it to focus solely on the relationship between note events.

The encoding and decoding processes both rely on F_{beat}. To encode a MIDI file, each note's absolute onset time is quantized and mapped to a relative Pos token within its corresponding measure. The decoding process is the reverse: the Decoder's output sequence, which contains only relative positional information, is rendered back into a standard MIDI file by using F_{beat} to restore absolute timings, tempo, and meter for each measure. This decoupled approach ensures that the generated output is not only musically coherent but also

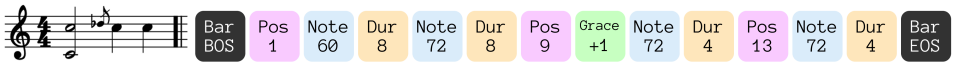

Fig. 3. An example of Tiny-REMI tokenization, where the sequence is delimited by `Bar` [BOS] and `Bar` [EOS] tokens, and each note event group (`Note` + `Dur`) is preceded by a `Pos` token marking its relative position.

structurally sound and easily convertible to standard musical formats like MIDI or MusicXML.

IV. Experiments

A. Dataset

We collected a dataset of approximately 7,700 pop song and piano cover audio pairs, primarily consisting of J-pop and K-pop. To ensure data quality, we filtered out pairs with a length difference greater than 30 seconds or a WP-std [5] greater than 1.0. The remaining pairs were then synchronized using the weakly-alignment method proposed in PiCoGen2 [3].[2] Our final training dataset consists of 4,752 pairs, totaling approximately 500 hours of audio. For evaluation, we curated a separate test set of 100 songs not seen during training, evenly distributed across four genres: C-pop, J-pop, K-pop, and Western pop.

B. Training Details

Our framework's two main components, the Extractor and the Decoder, are trained separately.

- **Extractor**: We adapt the AMT-APC [4] architecture. Unlike the original, which learns from multiple cover versions of a single song, we use one-to-one pairings. To encourage the model to produce a dense feature map rather than a sparse arrangement, we do not use style vectors and significantly lower the loss sampling parameter (θ^{matrix}). The model was trained for 10 epochs with a batch size of 2.
- **Decoder**: We use a GPT-NeoX [25] architecture with 8 Transformer layers, 8 attention heads, and a hidden size of 512, totaling approximately 25.5M parameters. The model was trained on sequences of up to 1,024 tokens. We used the AdamW optimizer [26] with a learning rate of 2×10^{-4}, scheduled with cosine annealing after a 10 epoch linear warmup. The model was trained for 100 epochs with a batch size of 128.

C. Baselines

We compare our model against several APCG baselines: PiCoGen2 [3], AMT-APC [4], and Music2MIDI [5]. Furthermore, we include the Human-performed piano cover as an upper bound for comparison. We also evaluate three versions of our own system:

- **Etude Extractor**: The output of our feature Extractor.
- **Etude - Default**: The Decoder's output using default, median style attributes.
- **Etude - Prompted**: The Decoder's output with manually selected style prompts for each song.[3]

D. Objective Metrics

Developing reproducible objective metrics for APCG is crucial. We propose three novel metrics designed to capture key dimensions of quality that correlate with human perception: similarity to the source, rhythmic fluency, and expressive dynamics.

- **Warp Path Deviation (WPD)** evaluates the structural similarity between the generated cover and the original song. This metric improves upon the WP-std from Music2MIDI [5] by calculating the standard deviation of the residuals from a linear regression of the DTW warp path. This approach makes WPD robust to global tempo variations while remaining sensitive to significant structural misalignments.
- **Rhythmic Grid Coherence (RGC)** is introduced to evaluate rhythmic fluency and detect timing jitter. It quantifies how well the inter-onset intervals (IOIs) of a performance adhere to a consistent underlying metrical grid by first identifying a base tempo unit (τ) from the most frequent IOIs, and then calculating the average deviation from the grid defined by τ. A lower RGC score indicates a more rhythmically precise and coherent performance.
- **IOI Pattern Entropy (IPE)** assesses the dynamic expressiveness and complexity of rhythmic patterns. This metric calculates the Shannon entropy of the distribution of overlapping n-grams within the generated performance's IOI sequence, after the IOIs have been converted into a series of discrete symbols via k-means clustering. A higher IPE score suggests more varied and less repetitive rhythmic motifs.

E. Subjective Evaluation

To complement our objective metrics, we conducted a subjective listening test with 101 volunteers, who were categorized into three groups based on their musical experience: amateurs ($<$3 years), intermediate (3-10 years), and expert ($>$10 years). Each participant was asked to listen to the original version of a song, followed by seven piano cover versions presented in a randomized and anonymous order. These versions included the human performance and the outputs from the six models listed in Section IV-C. To ensure fairness, all model-generated MIDI files and the human performance

[2]Our process modifies the original PiCoGen2 method. Instead of performing beat detection on both audio tracks, we detect beats [21] only on the source audio. We then find the corresponding timestamps in the cover audio using the MrMsDTW [23] implementation from the SyncToolbox Python package [24] before aligning at the **measure** level.

[3]While our model supports per-measure style injection, for this evaluation, a single set of style vectors was applied to the entire piece.

TABLE I

COMPREHENSIVE EVALUATION RESULTS OF HUMAN COMPOSERS, OUR PROPOSED ETUDE MODELS, AND OTHER BASELINES ON BOTH OBJECTIVE AND SUBJECTIVE METRICS. FOR OBJECTIVE METRICS, WE ALSO SHOW THE DELTA (Δ) FROM THE HUMAN PERFORMANCE. FOR SUBJECTIVE METRICS, HIGHER IS BETTER ($\uparrow$).

Model	**Objective Evaluation**			**Subjective Evaluation** ($\in$ [1, 5])			
	WPD	**RGC**	**IPE**	**SI** $\uparrow$	**FL** $\uparrow$	**DE** $\uparrow$	**OVL** $\uparrow$
Human	0.49	0.042	10.13	3.75 $\pm$ 1.10	4.03 $\pm$ 1.02	3.79 $\pm$ 1.06	3.92 $\pm$ 0.96
Etude - Default	0.21 (Δ 0.28)	0.020 (Δ 0.022)	9.02 (Δ 1.11)	3.16 $\pm$ 1.07	**3.73** $\pm$ 0.98	3.46 $\pm$ 1.05	**3.50** $\pm$ 0.99
Etude - Prompted	0.23 (Δ **0.26**)	0.026 (Δ 0.016)	9.11 (Δ 1.02)	3.17 $\pm$ 1.10	3.70 $\pm$ 1.05	**3.49** $\pm$ 1.06	3.46 $\pm$ 1.00
Etude Extractor	0.12 (Δ 0.37)	0.028 (Δ **0.014**)	10.62 (Δ **0.49**)	**3.41** $\pm$ 1.01	3.31 $\pm$ 1.13	3.35 $\pm$ 1.03	3.33 $\pm$ 1.00
PiCoGen2 [3]	1.00 (Δ 0.51)	0.059 (Δ 0.017)	7.97 (Δ 2.16)	2.88 $\pm$ 1.13	3.33 $\pm$ 1.12	2.73 $\pm$ 1.14	2.97 $\pm$ 1.04
AMT-APC [4]	0.09 (Δ 0.40)	0.114 (Δ 0.072)	10.69 (Δ 0.56)	2.64 $\pm$ 0.99	2.37 $\pm$ 1.11	2.71 $\pm$ 1.13	2.46 $\pm$ 1.04
Music2MIDI [5]	0.18 (Δ 0.31)	0.160 (Δ 0.118)	8.94 (Δ 1.19)	2.56 $\pm$ 1.06	2.29 $\pm$ 1.13	2.24 $\pm$ 1.09	2.27 $\pm$ 1.07

(which was first transcribed to MIDI) were synthesized into audio using the same piano synthesizer. All audio excerpts were 60-second clips extracted from the beginning of each song.

Following the methodology of PiCoGen2 [3], participants rated each cover on a 5-point Likert scale across four criteria:

- **Similarity (SI)**: How similar the cover is to the original song in melody, harmony, and rhythm.
- **Fluency (FL)**: The musical coherence and smoothness of the performance.[4]
- **Dynamic Expression (DE)**: How human-like and expressive the performance sounds.[5]
- **Overall (OVL)**: How much do the participants like the piano cover in the personal overall listening experience?

F. Results and Discussion

1) Evaluation Results: The comprehensive results for both objective and subjective evaluations are presented in Table I.

The objective metrics reveal that extreme scores are not always optimal; instead, the human performance provides a balanced target. For instance, in WPD, the lowest scores were achieved by models that perform overly literal transcriptions (Etude Extractor, AMT-APC). Similarly, for RGC, the Etude Decoder versions (Etude - Default, Etude - Prompted) achieved the lowest scores, indicating perfect adherence to a metrical grid, whereas the Human performance showed slightly higher deviation due to expressive timing. For IPE, the Human performance struck a balance between the monotonous patterns of PiCoGen2 (low IPE) and the chaotic randomness of AMT-APC (high IPE). This suggests that for objective metrics, closeness to the human benchmark is a strong indicator of a well-balanced and high-quality arrangement.

The subjective evaluation results in Table I confirm the superiority of our proposed framework. Statistical analysis reveals that all three Etude variants significantly outperform all three baseline models (PiCoGen2, AMT-APC, and Music2MIDI) in Overall Quality (OVL), with all planned pairwise comparisons yielding a corrected $p < 0.001$.

Diving into the details, the Human performance unsurprisingly achieved the highest scores across all criteria, establishing a clear upper bound. Among the AI models, both versions of our Etude Decoder were the top performers, with Etude - Default achieving the highest OVL score of 3.50. Notably, our decoder versions also lead in Fluency (FL) and Dynamic Expression (DE), indicating that their outputs are perceived as more natural and musical. Meanwhile, the Etude Extractor achieved the highest Similarity (SI) score among all models, confirming its effectiveness at capturing a dense and accurate representation of the source material. These results provide strong evidence that the Etude framework represents a substantial advancement in the quality of automatic piano cover generation.

TABLE II

DESCRIPTIVE STATISTICS OF THE OVERALL (OVL) SCORES FOR PARTICIPANT GROUPS BASED ON MUSICAL EXPERIENCE. SCORES ARE PRESENTED AS MEAN $\pm$ STANDARD DEVIATION.

Experience Group	**N**	**OVL Score**
Amateurs ($<$ 3 years)	30	3.32 $\pm$ 0.53
Intermediate (3-10 years)	39	3.09 $\pm$ 0.50
Expert ($>$ 10 years)	32	3.00 $\pm$ 0.60

2) Analysis of Rater Experience: To validate our subjective findings, we analyzed the OVL scores by rater experience (Table II). While a trend emerged where more experienced listeners gave harsher scores, a statistical test showed that the difference among the groups was not significant ($p = 0.070$). This suggests our main results are robust across listeners with varying musical expertise.

3) Ablation Study: Our proposed Etude framework introduces significant improvements over prior works in both its data processing pipeline and model architecture. To disentangle these effects and investigate the primary source of the performance gains, we conducted an ablation study focusing on the strongest baseline model, PiCoGen2.

To isolate the impact of the model architecture from the influence of the dataset, we first trained a new version of

[4]A low score may indicate issues such as rhythmic errors, missed notes, or awkward silences that disrupt the musical flow.

[5]A low score may suggest the performance is mechanical, lifeless, or emotionally flat.

TABLE III
SUBJECTIVE EVALUATION RESULTS FOR THE ABLATION STUDY, COMPARING THE EFFECTS OF THE DATASET AND MODEL ARCHITECTURE.

Model	SI ↑	FL ↑	DE ↑	OVL ↑
Etude - Default (Ours)	**3.38**	**3.53**	**3.58**	**3.55**
PiCoGen2* (Ours)	3.10	3.37	3.26	3.18
PiCoGen2 (Original)	2.97	3.19	2.63	2.98

the PiCoGen2 model, hereafter referred to as PiCoGen2*, following its official implementation but using our own high-quality dataset (detailed in Section IV-A). We then conducted a separate subjective listening test with 30 new participants. The experimental setup was identical to our main evaluation (Section IV-E), including the audio synthesis process and the four criteria (SI, FL, DE, OVL). The test compared the following three systems:

- **Etude - Default (Ours)**: Our proposed model trained on our proposed dataset.
- **PiCoGen2* (Ours)**: The baseline model retrained on our proposed dataset.
- **PiCoGen2 (Original)**: The official model trained on its original dataset.

The results of this ablation study are presented in Table III. The analysis reveals two key findings. First, while retraining the baseline on our dataset (PiCoGen2*) shows a slight improvement over the original PiCoGen2, the difference in OVL scores was not statistically significant ($p = 0.0656$). Second, and more crucially, our Etude - Default model significantly outperformed PiCoGen2* ($p = 0.0007$), even when both were trained on the exact same dataset. This provides strong evidence that while our data processing pipeline contributes positively, the primary driver for the substantial performance gain is the architectural innovations of the Etude framework itself.

V. CONCLUSION

In this paper, we addressed the key challenges of structural inconsistency and a lack of stylistic control in Automatic Piano Cover Generation (APCG). We proposed **Etude**, a novel three-stage framework built upon a rigorous data processing pipeline. Our approach modularizes the task into three distinct components—a Beat-Detector, an Extractor, and a Decoder—and introduces **Tiny-REMI**, a minimalist and efficient token representation. This decoupled design simplifies the learning process and, by leveraging a pre-extracted rhythmic framework, guarantees that the generated output is structurally consistent with the original song. Furthermore, by introducing a mechanism for injecting relative style attributes, our framework enables the generation of stylistically diverse and controllable piano covers.

Our main contributions include a highly extensible modular framework, an efficient Tiny-REMI representation that simplifies the learning task, and a novel style injection mechanism for controllable generation. Through extensive subjective evaluations, our Etude framework was shown to significantly outperform all state-of-the-art baselines, achieving a quality that approaches that of human arrangers.

For future work, we observe that the performance of our framework is fundamentally upper-bounded by the capabilities of its front-end components: the Beat-Detector and the Extractor. The framework's structural accuracy is limited by the precision of the beat tracker, while the Extractor's process of flattening the source audio into a single feature stream can lead to information loss. This limitation is evident in our subjective results (Table I), where both Etude Decoder versions scored markedly lower on Similarity (SI) than the direct Etude Extractor output. We hypothesize this is because the flattened features make it difficult for the model to identify the primary melody of the original song, causing the Decoder to generate incomplete melodic lines. Future work could thus explore both the integration of more advanced beat-tracking modules and the development of multi-stream extractors capable of disentangling different musical sources. Finally, our work highlights the persistent gap between current objective metrics and human perception. Developing learnable, perception-aligned evaluation models could provide a more reliable and automated benchmark for the field. Overall, the Etude framework provides a robust foundation for generating high-quality, controllable piano covers, paving the way for more musical and interactive AI-powered creative tools.

REFERENCES

[1] J. Choi and K. Lee, "Pop2piano: Pop audio-based piano cover generation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.

[2] C.-P. Tan, S.-H. Guan, and Y.-H. Yang, "Picogen: Generate piano covers with a two-stage approach," in *Proceedings of the 2024 International Conference on Multimedia Retrieval*, 2024, pp. 1180–1184.

[3] C.-P. Tan, H. Ai, Y.-H. Chang, S.-H. Guan, and Y.-H. Yang, "Picogen2: Piano cover generation with transfer learning approach and weakly aligned data," in *Proceedings of the 25th International Society for Music Information Retrieval Conference (ISMIR)*, San Francisco, CA, United States, Nov. 2024.

[4] K. Komiya and Y. Fukuhara, "Amt-apc: Automatic piano cover by fine-tuning an automatic music transcription model," *arXiv preprint arXiv:2409.14086*, 2024.

[5] T. Y. Yip and C.-j. Chau, "Music2midi: Pop music to midi piano cover generation," in *International Conference on Multimedia Modeling*, 2025, pp. 101–113.

[6] C.-Z. A. Huang, A. Vaswani, J. Uszkoreit, I. Simon, C. Hawthorne, N. Shazeer, A. M. Dai, M. D. Hoffman, M. Dinculescu, and D. Eck, "Music transformer," in *International Conference on Learning Representations*, 2019.

[7] S. Ji, X. Yang, and J. Luo, "A survey on deep learning for symbolic music generation: Representations, algorithms, evaluations, and challenges," *ACM Computing Surveys*, vol. 56, no. 1, pp. 1–39, 2023.

[8] G. Brunner, A. Konrad, Y. Wang, and R. Wattenhofer, "MIDI-VAE: Modeling dynamics and instrumentation of music with applications to style transfer," in *Proceedings of the 19th International Society for Music Information Retrieval Conference*, 2018, pp. 747–754.

[9] L.-C. Yang, S.-Y. Chou, and Y.-H. Yang, "MidiNet: A convolutional generative adversarial network for symbolic-domain music generation," in *Proceedings of the 18th International Society for Music Information Retrieval Conference*, 2017, pp. 324–331.

[10] 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*, vol. 32, no. 1, 2018.

[11] H.-M. Liu and Y.-H. Yang, "Lead sheet generation and arrangement by conditional generative adversarial network," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 722–727.

[12] 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*, 2020, pp. 1180–1188.

[13] 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*, vol. 35, no. 1, 2021, pp. 178–186.

[14] W. T. Lu, L. Su *et al.*, "Transferring the style of homophonic music using recurrent neural networks and autoregressive model." in *ISMIR*, 2018, pp. 740–746.

[15] G. Hadjeres, F. Pachet, and F. Nielsen, "Deepbach: a steerable model for bach chorales generation," in *International conference on machine learning*. PMLR, 2017, pp. 1362–1371.

[16] 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)*, 2018, pp. 786–793.

[17] 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*, vol. 31, pp. 1953–1967, 2023.

[18] H. H. Tan and D. Herremans, "Music fadernets: Controllable music generation based on high-level features via low-level feature modelling," in *Proc. of the International Society for Music Information Retrieval Conference*, 2020.

[19] C. Donahue, J. Thickstun, and P. Liang, "Melody transcription via generative pre-training," in *ISMIR*, 2022.

[20] K. Toyama, T. Akama, Y. Ikemiya, Y. Takida, W. Liao, and Y. Mitsufuji, "Automatic piano transcription with hierarchical frequency-time transformer," in *Ismir 2023 Hybrid Conference*, 2023.

[21] J. Zhao, G. Xia, and Y. Wang, "Beat transformer: Demixed beat and downbeat tracking with dilated self-attention," in *Ismir 2022 Hybrid Conference*, 2022.

[22] 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)*. IEEE, 2023, pp. 1–5.

[23] T. Prätzlich, J. Driedger, and M. Müller, "Memory-restricted multiscale dynamic time warping," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 569–573.

[24] M. Müller, Y. Özer, M. Krause, T. Prätzlich, and J. Driedger, "Sync toolbox: A python package for efficient, robust, and accurate music synchronization," *Journal of Open Source Software*, vol. 6, no. 64, p. 3434, 2021.

[25] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang *et al.*, "GPT-NeoX-20B: An open-source autoregressive language model," in *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, May 2022, pp. 95–136.

[26] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2019.