

HiTaiko: Hierarchical osu!taiko Chart Generation via Metrical Candidate Selection

Sungjung Park

Department of Applied Statistics, Chung-Ang University
Seoul, Republic of Korea
woosou2000@gmail.com

Abstract

We present HiTaiko, an automatic osu!taiko chart-generation system developed for the MIREX 2026 Rhythm Game Chart Generation task. HiTaiko formulates tap placement as selection over a dense 1/24-beat metrical candidate grid derived from the provided uninherited timing information. Local log-Mel audio features and metrical embeddings are contextualized by a Transformer to produce a shared placement salience for each candidate. Five difficulty levels are then generated using learned ordered thresholds, yielding an explicitly nested hierarchy across tiers. This design is motivated by the strong monotonic structure observed in the curated clean269 corpus, where 90.84% of positive metrical positions follow a monotonic five-tier inclusion pattern. After placement, separate modules predict note size and Don/Kat attributes and generate DENDEN and DRUMROLL objects, followed by conflict resolution and hierarchy-preserving post-processing. The submitted system is fully frozen at inference time and outputs all five osu!taiko difficulty charts from organizer-provided audio, timing information, and beatmap templates.

1. Introduction

Automatic rhythm-game chart generation requires a system to align player actions with musical structure while producing charts with clearly differentiated difficulty levels. In osu!taiko, this involves deciding where tap notes should occur, assigning note attributes such as Don/Kat and normal/big size, and generating special objects such as drumrolls and denden. For the MIREX 2026 Rhythm Game Chart Generation task, the ground-truth uninherited timing information is provided, allowing chart generation to focus on the placement and characterization of playable hit objects rather than beat tracking.

We present HiTaiko, an automatic osu!taiko chart-generation system based on metrical candidate selection and hierarchical difficulty generation. Instead of predicting arbitrary note timestamps directly in continuous time, HiTaiko constructs an exact 1/24-beat candidate grid from the provided timing map. For each candidate, a convolutional audio encoder extracts information from a local log-Mel spectrogram segment, while metrical position is represented by within-beat tick and rhythmic-subdivision embeddings. A Transformer encoder then contextualizes the sequence of candidates and produces a shared placement salience score for each position.

The five difficulty levels are generated from this shared salience representation using learned ordered thresholds. Higher difficulty tiers therefore retain note positions selected at lower tiers while progressively adding additional candidates. This formulation separates estimation of rhythmic importance from difficulty selection and provides a direct mechanism for generating progressively denser charts while preserving common rhythmic anchors.

After note placement, HiTaiko uses separate models to determine note size and Don/Kat color, followed by dedicated inference modules for DENDEN and DRUMROLL events. A

final post-processing stage resolves conflicts between taps and special objects and restores the ordered difficulty hierarchy before the generated hit objects are inserted into the organizer-provided osu!taiko templates.

2. Dataset and Training Data

HiTaiko uses a curated five-difficulty corpus for its main chart-generation components and a larger, separately constructed corpus for drumroll modeling. The primary corpus, referred to as clean269, contains 269 osu!taiko beatmapsets and 1,345 selected charts. Each song contains exactly one chart corresponding to each difficulty tier T_1 – T_5 . For all 269 songs, the five selected charts share an identical uninherited timing map, providing a common metrical reference across difficulty levels.

For placement modeling, the five charts of each song were aligned within a common metrical candidate space. A full 1/24-beat grid was generated from the uninherited timing information, producing 2,779,310 candidate positions across the 269 songs. Note positions from the five human-authored difficulty charts were mapped to their nearest candidate within a tolerance of ± 5 ms. Among 246,092 unique note positions observed across difficulties, 246,028 were represented on the 1/24-beat grid, corresponding to a coverage of 99.974%. Only 64 positions remained outside the specified tolerance.

Each candidate is associated with five binary labels indicating whether that metrical position contains a note in $T_1, \dots, T_5$. Although the human-authored charts are not strictly nested, 90.84% of positive metrical positions follow a monotonic five-tier inclusion pattern, where a note introduced at a given tier is retained through all higher tiers. This strong empirical tendency motivated the ordered hierarchical formulation used by HiTaiko.

The clean269 collection also provides the human tap-note annotations used by the subsequent note-attribute components;

across its 1,345 charts, it contains 733,581 tap notes.

Drumroll modeling uses a larger, separately prepared corpus because drumroll annotations are substantially sparser. Starting from 727 `osu!taiko` beatmapsets, two sets were removed during cleaning, resulting in `clean725`. Exact duplicate audio files were grouped using audio hashes into shared AudioUIDs, yielding 712 unique AudioUIDs. The final drumroll onset model was refit on all 712 AudioUIDs, comprising 1,927,222 onset candidates. The duration model was trained separately on 733 labeled drumroll intervals drawn from 197 positive AudioUIDs.

3. System Architecture

3.1 Metrical Candidate Construction

HiTaiko formulates tap placement over a discrete set of metrical candidates rather than over arbitrary continuous timestamps. The candidate grid is constructed directly from the provided uninherited timing points. When timing information from multiple charts is available for a beatmapset, the system groups charts by their uninherited timing signatures and selects the most frequently occurring signature as the representative timing map.

For each timing segment with beat length b , candidate positions are generated at an interval of

$$\Delta t = \frac{b}{24}, \quad (1)$$

forming an exact 1/24-beat grid. At a timing change, the grid is restarted from the new uninherited timing point using the new beat length. Positions preceding the first timing point are generated by backward extrapolation using the first beat length so that the candidate representation covers the full audio timeline. Numerical duplicates that may occur at timing-segment boundaries are removed.

Each candidate is assigned a 24-way within-beat position and a rhythmic-subdivision category corresponding to the smallest conventional subdivision on which that tick lies. The eight categories are 1/1, 1/2, 1/3, 1/4, 1/6, 1/8, 1/12, and 1/24. The same representation is used during training and inference.

3.2 Placement Model

For each metrical candidate, HiTaiko extracts a local audio representation from an 80-bin log-Mel spectrogram computed at 22.05 kHz using a 1,024-sample FFT and a hop size of 256 samples. The local context is defined relative to the current beat length: 32 spectrogram positions are sampled uniformly from 0.5 beat before to 0.5 beat after the candidate. Log-Mel values are clipped to an 80-dB range and normalized to $[0, 1]$.

Each local 80×32 audio patch is processed by a convolutional encoder with three 3×3 convolutional blocks containing 16, 32, and 64 channels. Pooling is applied along the frequency dimension while preserving local temporal resolution. Adaptive pooling and a linear projection produce a 128-dimensional audio feature for each candidate.

Metrical information is encoded using an 8-dimensional embedding of the 24 within-beat tick positions and a 4-dimensional embedding of the eight rhythmic-subdivision categories. The 128-dimensional audio representation and the two metrical embeddings are concatenated and projected to a 128-dimensional token representation.

Candidate tokens are then processed by a two-layer Transformer encoder with four attention heads and a 512-dimensional feed-forward layer. Sinusoidal positional encoding is added before the Transformer. During inference, candidates are processed in overlapping sequences of 192 positions, consisting of 48 candidates of left context, 96 target candidates, and 48 candidates of right context. Only predictions for the central 96 positions are retained.

Finally, a shared linear head maps each contextualized candidate representation to a scalar placement salience. This salience is shared across all five difficulty levels.

3.3 Hierarchical Difficulty Generation

HiTaiko converts each shared salience score into five difficulty-specific placement decisions using learned ordered thresholds. The threshold module contains one base parameter and four learnable gap parameters. Each gap is transformed using

$$\delta_j = \text{softplus}(g_j) + 10^{-4}, \quad (2)$$

ensuring strictly positive gaps. Cumulative addition produces thresholds satisfying

$$\tau_1 > \tau_2 > \tau_3 > \tau_4 > \tau_5, \quad (3)$$

where T_1 is the lowest difficulty and T_5 the highest. This constraint favors consistent difficulty progression over reproducing the minority of non-monotonic transitions observed in the training charts.

For candidate i with salience s_i , the logit for tier t is

$$z_{i,t} = s_i - \tau_t. \quad (4)$$

At inference time, a candidate is selected when the corresponding sigmoid probability is at least 0.5, which is equivalent to

$$s_i \geq \tau_t. \quad (5)$$

Because the thresholds are ordered, the model directly guarantees

$$T_1 \subseteq T_2 \subseteq T_3 \subseteq T_4 \subseteq T_5. \quad (6)$$

After threshold-based selection, HiTaiko applies a 30-ms minimum tap inter-onset-interval constraint. The cleanup is hierarchy-aware: lower-tier notes already retained are protected, while newly introduced higher-tier candidates may be removed. Weighted interval scheduling using placement probability as the weight selects the highest-scoring spacing-valid subset.

3.4 Note Attribute Prediction

After tap positions have been selected, HiTaiko assigns size (normal or big) and color (Don or Kat) attributes. Both models operate only on the generated tap sequence.

For size prediction, each tap is represented using its difficulty tier, 24-way within-beat position, rhythmic-subdivision category, and sequence-context features describing normalized log inter-onset intervals to the previous and next taps together with neighbor-existence indicators. A structural MLP first produces a base score. A second model adds an audio residual derived from a local beat-relative log-Mel patch.

Final BIG assignment uses a difficulty-specific BIG prior learned from clean269 rather than an independent 0.5 threshold. For a tier containing N taps,

$$K = \text{round}(Nr_t), \quad (7)$$

where r_t is the stored BIG rate for tier t . The K highest-scoring taps are assigned as big notes.

Don/Kat prediction uses a related structural-plus-audio architecture. Its structural component receives the difficulty tier, within-beat tick, and previous/next inter-onset context features. A local-audio residual model produces an initial Don/Kat logit. These logits are subsequently refined by a single-layer GRU history model operating over the chronological tap sequence. The final sigmoid output is classified as Kat at a threshold of 0.5 and as Don otherwise.

The resulting combinations correspond to the four ordinary tap types: DON, KAT, BIG_DON, and BIG_KAT.

3.5 Special Objects and Final Post-processing

HiTaiko models DENDEN and DRUMROLL separately from ordinary tap placement because these objects require both an onset and a duration. Both modules operate on a coarser 1/4-beat candidate grid.

For DENDEN, the start model is structure-only and ranks 1/4-beat candidates independently for each difficulty tier. The decoder selects one valid candidate per tier. Duration is predicted by a separate structure-plus-audio model using forward context from the selected start position, and the predicted duration is quantized to 0.25-beat increments.

DRUMROLL onset localization is performed on a 1/4-beat grid using a dedicated audio-based localization model. Local onset peaks are ranked, and proposals exceeding the frozen deployment threshold are paired with predictions from a separate duration model. Drumroll durations are constrained to 0.5–8 beats and quantized to 0.25-beat increments. Non-overlapping proposals are selected greedily in descending onset confidence, with at most six drumrolls retained per song. The resulting intervals are shared across the five difficulty tiers.

The final merge stage resolves interactions between taps and special objects. DENDEN has priority over DRUMROLL: when their active intervals overlap within a tier, the conflicting drumroll is removed. Ordinary taps within protected windows around retained special objects are also removed.

Because special-object filtering can remove different taps from different difficulty levels, the ordered tap hierarchy is restored after merging. Lower-tier taps that are no longer present in the next higher tier are removed, re-establishing

$$T_1 \subseteq T_2 \subseteq T_3 \subseteq T_4 \subseteq T_5. \quad (8)$$

DENDEN events are exported as osu!taiko spinners and DRUMROLL events as duration-aware sliders.

4. Training Procedure

For components trained on clean269, model selection was performed using a fixed five-fold song-level split, ensuring that charts from the same song did not appear in both training and validation folds. A common epoch was selected across all five folds according to the corresponding validation metric, and that

training budget was subsequently used to refit the model on the complete training corpus.

For the placement model, each fold was trained for 12 epochs and evaluated using pooled MacroAP across the five difficulty tiers. The highest mean validation MacroAP across the five folds occurred at epoch 10, which was used for the final full-data refit. Training uses binary cross-entropy over the five ordered tier outputs and computes loss only for the central 96 target candidates of each 192-candidate sequence.

For size, five-fold validation selected 29 epochs for the structural model and 5 epochs for the audio-residual model, using Average Precision as the primary criterion for the imbalanced BIG/normal task. For Don/Kat, the final full-data training budgets were 11 epochs for the structural model, 4 epochs for the audio residual, and 29 epochs for the history GRU.

Special-object models were trained independently. The structure-only DENDEN onset model was selected by five-fold MacroAP, with epoch 19 used for the final model. DENDEN duration used 41 epochs for the structural model and 24 epochs for the forward-audio residual, followed by full-data refitting on all 1,720 labeled DENDEN events.

For DRUMROLL, the selected global training budgets were 7 epochs for onset localization and 23 epochs for duration prediction. These budgets were used to refit the onset model on all 712 AudioUIDs and the duration model on all 733 labeled intervals. The final onset deployment threshold was set to 0.10986328 and frozen together with the remaining decoding parameters.

No training, adaptation, or parameter tuning is performed on MIREX test songs.

5. Inference and Output Generation

At inference time, HiTaiko receives an organizer-provided .osz template containing five osu!taiko difficulty files, the song audio, and ground-truth uninherited timing points. The frozen entry point verifies that the five standard difficulty tiers are present, reference the same audio, and contain an identical uninherited timing map. Existing template HitObjects are not used as model inputs.

Generation proceeds through four frozen stages. First, the base generator constructs the 1/24-beat candidate grid, predicts ordered T_1 – T_5 tap placements, applies the minimum-IOI cleanup, and predicts size and Don/Kat attributes. Second, the DENDEN and DRUMROLL modules generate special-object proposals. Third, the final merge stage resolves conflicts and restores the ordered tap hierarchy. Finally, the generated HitObjects for each tier are inserted into the corresponding organizer-provided .osu template.

The five populated difficulty files are then packaged into the final .osz output after integrity and tier-coverage checks.

6. Conclusion

We presented HiTaiko, an automatic osu!taiko chart-generation system that formulates placement as selection over a dense 1/24-beat metrical grid and generates five nested difficulty tiers using a shared salience representation and learned ordered thresholds. The frozen system further predicts tap attributes and special objects and generates all five charts from organizer-provided audio, timing information, and beatmap templates.