

## A Technical Appendices and Supplementary Material

### A.1 Model Architecture

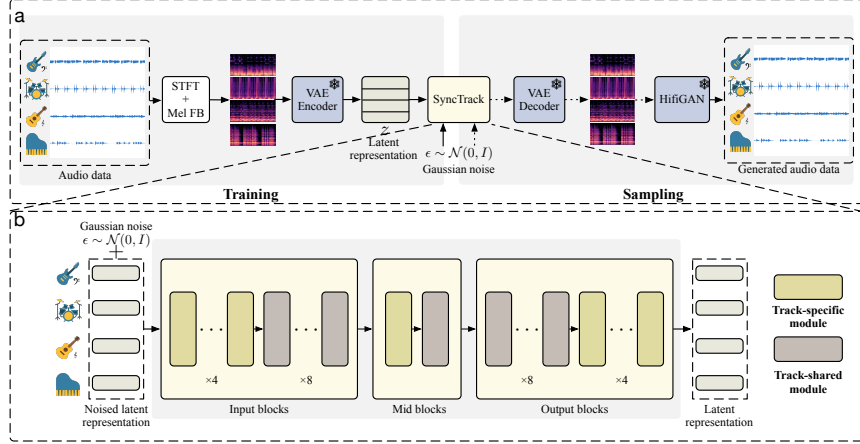


Figure 2: Detailed SyncTrack Architecture

Figure 2 presents an overview of the SyncTrack architecture, highlighting the main data flow and core modules during both training and sampling.

Fig 2a shows the end-to-end pipeline. Raw multi-track audio is first converted to Mel-spectrograms via STFT and Mel filter banks, then compressed into latent representations using a VAE encoder. SyncTrack operates in this latent space, taking as input both noisy latent variables and learnable instrument priors, which are initialized as one-hot vectors and trained as embeddings to capture track-specific features.

Fig 2b details the SyncTrack backbone—a single U-Net model with shared parameters across all tracks. All tracks were processed jointly in a parameter-sharing manner, enabling information sharing and efficient modeling of track-wise dependencies. As shown in Fig 3a, each track-shared module contained residual modules, inner-track attention, and two specialized cross-track attention modules. The Cross-Track Attention aggregates information globally across all tracks and time-frequency bins, while the Time Specific Cross-Track Attention focuses on synchronizing features across tracks at each time step, thus improving beat alignment. Learnable instrument priors are injected into the network to preserve track-specific information within the shared framework.

Fig 3c illustrates the two cross-track attention mechanisms: Cross-Track Attention coordinates tracks globally over the full time-frequency space, and Time-specific Cross-Track Attention aligns tracks locally at each frame for precise synchronization.

During sampling, output latents were decoded into Mel-spectrograms by the VAE decoder and then converted to multi-track waveforms with a HiFi-GAN vocoder. This modular design ensured both efficient training and high-fidelity generation, while the integration of cross-track attention and instrument priors enabled SyncTrack to model both global and local inter-track dependencies effectively.

### A.2 Beat Tracking Implementation Details

**Tools and Default Settings.** We utilize the *RNNDOWNBeatProcessor* and *DBNDOWNBeatTrackingProcessor* from the madmom [Böck et al., 2016] library for beat extraction throughout our experiments. Unless otherwise specified, we set the frame rate (fps) to 150Hz and used madmom’s default transition lambda (tl) value [Böck et al., 2016] for all main results.

**Key Parameters.** Beat detection accuracy directly affects rhythm-related metrics. To examine the robustness of our metrics, we systematically varies two key parameters in beat tracking:

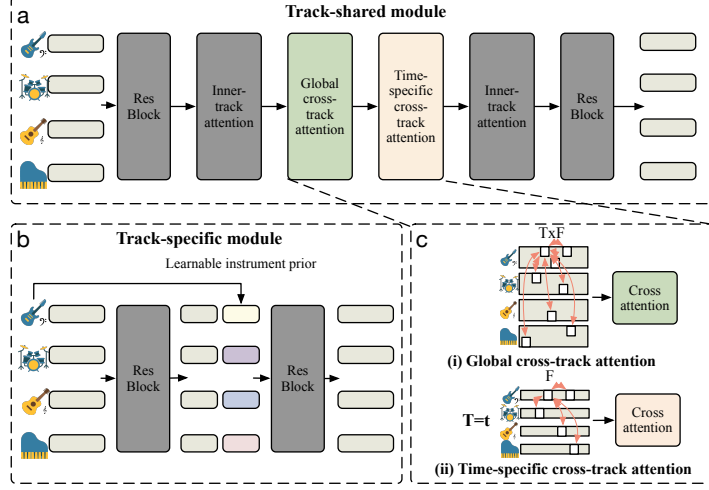


Figure 3: Illustration of the (a) track-shared module and (b) track-specific module. In (a), we leverage inner-track attention to capture the inner-track rhythmic stability and devise (c) two cross-track attention submodules to capture cross-track rhythmic stability and synchronization. In (b), we construct a learnable instrument prior to capture timbre and other track-specific features.

- *Frames per Second (fps)*: Higher fps provides finer temporal resolution, enabling more precise beat localization—especially important for multi-track or sparsely rhythmic content. Lower fps reduces computational cost but may degrade detection accuracy.
- *Transition Lambda (tl)*: This parameter controls temporal smoothing during beat sequence inference. Higher tl enforces smoother, more consistent tempo estimation, but may mask genuine rhythmic instabilities in generated tracks. Excessively high values can artificially inflate rhythmic stability scores by obscuring local irregularities. We find the default tl offered a good balance between sensitivity and smoothing, reliably reflecting true rhythmic quality.

### A.3 Parameter Sensitivity Analysis

To rigorously assess the robustness of our evaluation metrics, we conduct experiments across a grid of beat tracking configurations, varying both fps and tl. Figures 4 and 5 as well as Table 4 summarizes the effects of these parameters on our three metrics: Inner-track Rhythmic Stability (IRS), Cross-track Beat Synchronization (CBS), and Cross-track Beat Dispersion (CBD).

Across all settings, the relative ranking of models remains stable, demonstrating that our metrics are robust to changes in beat tracking hyperparameters.

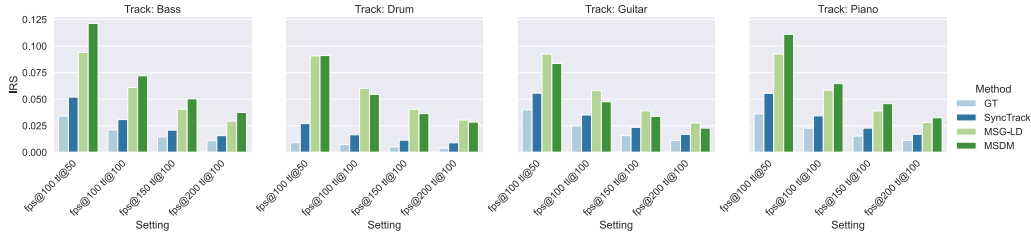


Figure 4: Comparison of IRS across hyperparameter settings,

**Inner-track Rhythmic Stability (IRS).** As shown in Figure 4, the relative ordering among GT, SyncTrack, MSG-LD, and MSDM remains unchanged as beat tracking parameters vary. This demonstrates that our rhythmic stability metric is robust to beat tracking hyperparameters, and that model comparisons are reliable regardless of the specific configuration.

Besides, increasing either fps or tl results in a consistent decrease in IRS values across all methods. This trend reflects the influence of beat tracking hyperparameters: higher fps yield finer temporal resolution, allowing for more precise and stable beat localization, while higher tl enforces greater temporal smoothing and more regular tempo estimation. Both effects reduce the measured variance of beat intervals, thereby lowering IRS scores.

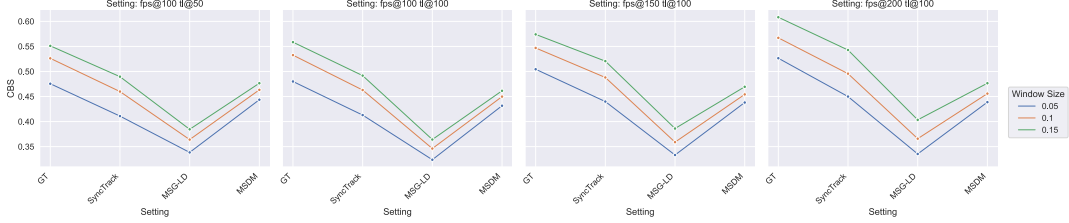


Figure 5: Comparison of CBS across hyperparameter settings.

**Cross-track Beat Synchronization (CBS).** As shown in Figure 5, CBS exhibits a consistent dependency on both beat tracking hyperparameters and the chosen window size. Varying the window size has a clear and intuitive effect: smaller window sizes impose a stricter criterion for considering beats as synchronous, leading to lower CBS scores, whereas larger window sizes relax this criterion and yield higher CBS values. This relationship holds consistently for all methods and parameter settings.

Despite these variations in absolute CBS values, the relative ranking among GT, SyncTrack, MSG-LD, and MSDM remains unchanged across all configurations. Ground truth consistently achieves the highest CBS, and the ordering among models was preserved regardless of the hyperparameter or window size choices. This stability demonstrates the robustness of CBS as a comparative metric for evaluating model performance under diverse evaluation settings.

**Cross-track Beat Dispersion (CBD).** Table 4 presents detailed statistics for CBD mean, std, and median. Moreover, although the absolute values of CBD metrics varies slightly with fps and tl, the relative model ranking is almost unaffected. This further corroborates the insensitivity of our evaluation framework to beat-tracking hyperparameters.

Table 4: Comparison of CBD across hyperparameter settings.

Setting	Metrics	Ground Truth	SyncTrack	MSG-LD	MSDM
fps@100 tl@50	CBD (mean) ↓	0.2206	<b>0.2589</b>	0.3644	0.2923
	CBD (std) ↓	0.1693	<b>0.2241</b>	0.2822	0.2370
	CBD (median) ↓	0.1769	<b>0.2058</b>	0.3350	0.2424
fps@100 tl@100	CBD (mean) ↓	0.2143	<b>0.2522</b>	0.3829	0.3205
	CBD (std) ↓	0.1556	<b>0.2101</b>	0.2708	0.2329
	CBD (median) ↓	0.1762	<b>0.2060</b>	0.3679	0.2870
fps@150 tl@100	CBD (mean) ↓	0.2412	<b>0.2681</b>	0.3714	0.3127
	CBD (std) ↓	0.1578	<b>0.2131</b>	0.2642	0.2217
	CBD (median) ↓	0.2066	<b>0.2258</b>	0.3545	0.2811
fps@200 tl@100	CBD (mean) ↓	0.2642	<b>0.2926</b>	0.3590	0.3109
	CBD (std) ↓	0.1639	0.2203	0.2534	<b>0.2134</b>
	CBD (median) ↓	0.2316	<b>0.2545</b>	0.3407	0.2807

#### A.4 Case Study of Rhythm Evaluation Metrics

To further validate the effectiveness and interpretability of our proposed rhythm evaluation metrics, we present a case study comparing both Ground Truth (GT) samples from the Slakh2100 dataset and generated samples from baseline models. We focus on three key metrics: IRS, CBS and CBD.

**Inner-track Rhythmic Stability.** Figure 6 shows two representative drum tracks from the GT dataset (left) and two from baseline-generated samples (right). For each, we visualize the spectrogram, beat

Table 5: Comparison of CBS and CBD for ground truth and generated samples.

	Sample	CBD(mean)	CBD(std)	CBD(median)	CBS
Slakh2100	①	0.0702	0.0481	0.0709	0.5286
	②	0.0628	0.0677	0.0553	0.7857
Generated from baselines	③	0.2619	0.2512	0.1522	0.3629
	④	0.3975	0.2387	0.3815	0.3041

and downbeat activations, and the extracted beat sequence. The GT drum tracks exhibit highly regular and stable beat intervals, as reflected in both the visualized beat grid and the very low IRS values (0.0049 and 0.0051). In contrast, the baseline-generated tracks display irregular beat sequences, with beat intervals that fluctuate over time and substantially higher IRS values (0.1859 and 0.1494). These results demonstrate that IRS effectively captures the stability of rhythmic patterns within a single track and aligned with intuitive human judgments.

**Multi-track Beat Synchronization.** We further analyze multi-track synchronization using two Ground Truth examples and two baseline-generated examples, as shown in Figure 7 and Table 5. The Ground Truth samples exhibit strong cross-track synchronization: beats from different instruments (bass, drums, guitar, and piano) are well aligned, as visible in the vertical alignment of beat annotations across tracks. Correspondingly, both CBS and CBD metrics indicate high synchronization and low dispersion across different track rhythm.

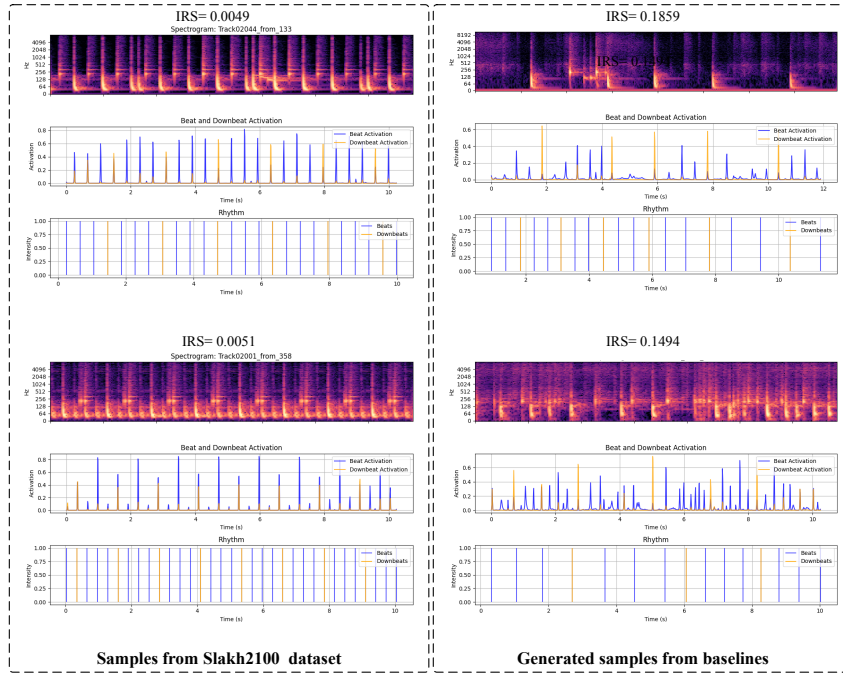


Figure 6: Case study of IRS for ground truth and generated samples.

In contrast, the generated samples display chaotic beats across tracks, and in some cases, entire tracks are empty. This is reflected in the lower CBS values and higher CBD values, indicating both worse cross-track alignment and larger rhythmic dispersion between tracks.

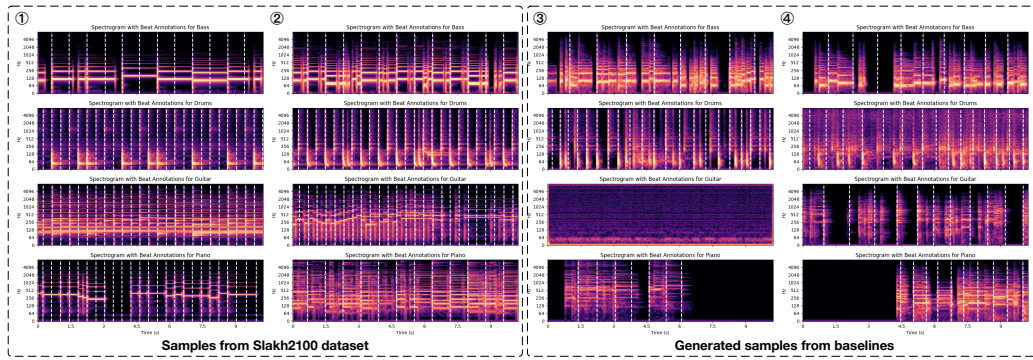


Figure 7: Visualization of cross-track synchronization in ground truth and generated samples.