MUSEDUO: UNIFIED AUDIO-MIDI MUSIC GENERATION, SEPARATION, TRANSCRIPTION, RENDERING, AND EDITING

Anonymous authors
Paper under double-blind review

ABSTRACT

Music generation has developed as two largely separate fields: *symbolic* generation, which produces MIDI or scores, and *audio* generation, which models waveforms. Musicians work across both, composing and arranging in symbolic terms and working directly with timbre and texture in the audio domain. We present **MUSEDUO**, a diffusion/flow transformer (DiT) that 1) is trained on both music mixes and multi-track instrument stems, and 2) jointly models audio and MIDI, enabling a broad range of music tasks. MUSEDUO comprises three key mechanisms: (i) a unified audio-MIDI multimodal latent rectified-flow architecture with a novel MIDI-VAE, (ii) a concise training recipe that realizes diverse tasks via systematic inference-time configuration of input, output, and conditioning, and (iii) on-the-fly MIDI extraction that alleviates the need for costly annotation. We fine-tune a pretrained text-to-audio music DiT, and evaluate on 10 music tasks spanning single-track mix generation, multi-track (stem) generation, source separation, single-track transcription, multi-track transcription, timbre transfer, MIDI-to-audio rendering, multi-track text-to-MIDI generation, and multiple note-level note editing tasks. With a unified design, MUSEDUO matches or outperforms 12 recent task-specific baselines on the majority of evaluated tasks and metrics.

1 INTRODUCTION

Music production spans two complementary representations of music. In the symbolic domain such as the Musical Instrument Digital Interface (MIDI) protocol, a piece is a set of discrete note events, pitches, onsets, durations which are then orchestrated, arranged, edited, and either performed or rendered to the final output. In the audio domain, it is a continuous waveform which carries the timbre and texture a score leaves unspecified. We posit that music generation methods should be aligned to operate in both domains jointly rather than in either one alone.

Generative modeling has advanced rapidly in both domains, from raw waveform synthesis to text-conditioned generation in the audio domain (van den Oord et al., 2016; Dhariwal et al., 2020; Evans et al., 2025), and multi-track symbolic generation in the score domain (Dong et al., 2018; Wu et al., 2026a) (full survey in §2). Each domain, however, remains mostly confined to its own output. Even the closest prior attempt at bridging them, U-MusT (Jung et al., 2026), trains separate checkpoints per direction; to the best of our knowledge, no existing model jointly models audio and MIDI via a single, unified generative process.

We propose MUSEDUO, a single latent diffusion/flow model trained jointly on audio and MIDI to achieve a wide variety of tasks as shown in Fig. 1 and Table 1. We do so via fine-tuning a pretrained text-to-music (TTM) generation model with symbolic data (MIDI) aligned to the existing audio latents via a new MIDI variational autoencoder (MIDI-VAE). During fine-tuning, we employ multi-task synthetic instruct-pair data creation using on-the-fly audio-to-MIDI extraction that alleviates the need for ground-truth MIDI, assign audio and MIDI latents independent noise levels so either can serve as a condition or a generation target at inference, also add a time-aligned reference audio condition via channel concatenation enabling source separation, accompaniment generation, inpainting, and note-level editing. We evaluate MUSEDUO on 10 tasks: single-track mix generation, multi-track (stem) generation, source separation, single-track transcription, multi-track transcription, tim-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

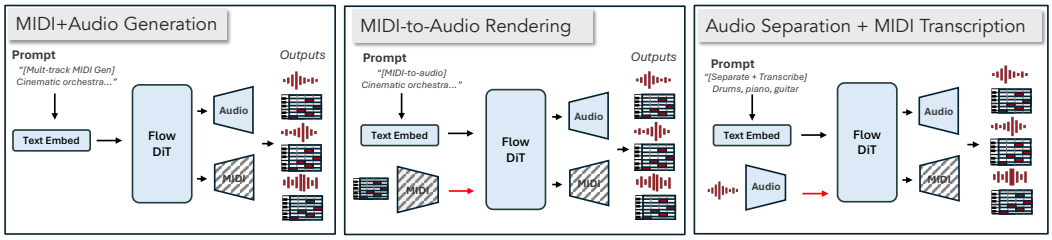


Figure 1: **MUSEDUO Inference.** Our approach jointly models audio and MIDI together to achieve a wide variety of powerful music applications at inference-time. (Left) Joint text-to-audio and MIDI generation with either single-track '*mix*' or multi-track '*stems*' outputs. (Middle) MIDI-to-Audio rendering that inputs MIDI and outputs high-quality audio performance with open-vocabulary instrument control. (Right) Single-pass separation and MIDI-transcription with open-vocabulary instrument control. Additionally, MUSEDUO is also capable of instrumental removal, accompaniment generation, audio-to-audio instrument timbre transfer, and more precise note-level editing tasks.

bre transfer, MIDI-to-audio rendering, multi-track text-to-MIDI generation, and multiple note-level note editing tasks, each of which has otherwise required a dedicated, task-specific system.

Our contributions can be summarized as follows:

- A unified audio-MIDI latent diffusion/flow architecture that jointly models time-aligned audio and MIDI, with a novel MIDI-VAE that maps discrete MIDI into continuous latents, enabling a single generative process for both modalities.
- A concise multimodal training recipe with one objective, combining scalable on-the-fly MIDI extraction with noise scheduling that supports both joint and cross-modal conditionals, realizing diverse music tasks through inference-time configuration of latent roles and conditioning.
- Comprehensive evaluation over 10 tasks, including generation, separation, transcription, rendering, and editing, demonstrating that MUSEDUO matches or outperforms 12 recent specialist baselines on the majority of evaluated tasks and metrics.

Audio samples across applications at `https://museduo-demo.vercel.app/`.

2 RELATED WORK

2.1 MUSIC GENERATION

Audio-domain. Neural audio generation has progressed through autoregressive, adversarial, and diffusion models over raw waveforms (van den Oord et al., 2016; Mehri et al., 2017; Donahue et al., 2019; Engel et al., 2019; Kong et al., 2021; Chen et al., 2021), converging on the latent representations that later text-conditioned, transformer-based (Dhariwal et al., 2020; Borsos et al., 2023; Agostinelli et al., 2023; Copet et al., 2023) and diffusion-based (Schneider et al., 2024; Liu et al., 2023a; Chen et al., 2024; Evans et al., 2024; 2025) systems inherit, including recent hybrid autoregressive-diffusion variants (Gong et al., 2026; Yang et al., 2025; Lee et al., 2026) that decode audio from learned, non-symbolic intermediate embeddings.

Multi-stem systems generate instrument-separated outputs (Parker et al., 2024; Rouard et al., 2025; Yao et al., 2025; Karchkhadze et al., 2024; Postolache et al., 2024), but largely for a fixed instrument set or one stem at a time. Stemphonic (Wu et al., 2026b) instead generates a variable-sized instrument pack in a single pass via noise-sharing; MUSEDUO builds on Stemphonic and additionally generates the corresponding MIDI tracks.

Symbolic-domain. Symbolic models operate on note and/or event representations of music (e.g., piano roll, guitar tabs, MIDI), using relative self-attention (Huang et al., 2019), beat-based tokenization (Huang & Yang, 2020), and piano-roll or lead-sheet diffusion (Min et al., 2023; Wang et al., 2024) for coherent, long-form generation. Multi-track work enables multi-instrument compositions (Dong et al., 2018; Yu et al., 2022; Thickstun et al., 2024; Qu et al., 2025; Wang et al., 2025), and text-to-MIDI follows user instructions (Lu et al., 2023; Wu et al., 2026a), but all still

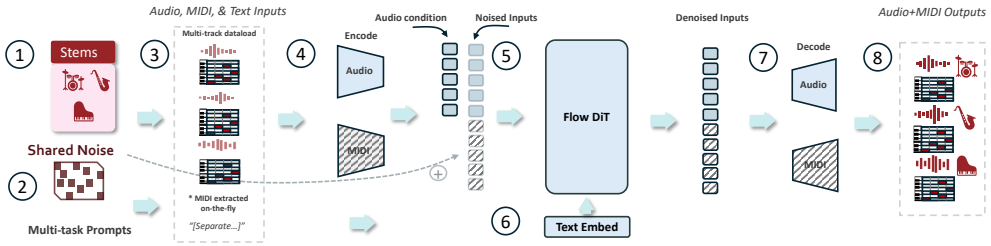


Figure 2: **MUSEDUO Training and Pipeline.** We (1) load a batch of stem packs per song (2) assign shared noise per stem pack, following (Wu et al., 2026b) (3) preprocess and produce a batch of multi-task multi-modal input-output pairs where per-stem MIDI sequences are extracted on-the-fly via music and drum transcription (4) encode all (audio, MIDI) pairs into latents using separate audio and MIDI VAEs (5) fuse audio and MIDI latents before our DiT backbone and apply diffusion/flow noise (6) encode text prompts as embeddings (7) denoise the MIDI and audio latents and compute training loss (8) optionally decode latents to usable audio and/or MIDI files.

need synthesizers to hear. Without need of synthesizers, MUSEDUO generates time-aligned MIDI and audio jointly, following a text prompt and a variable set of instruments in one inference pass.

Joint audio and symbolic domain. A growing body of work generates multiple modalities jointly, but pairs that need only agree semantically (Ruan et al., 2023; Kim et al., 2024; Bao et al., 2023; Wang et al., 2026), unlike audio and MIDI, which must align at the level of individual note onsets. Existing work bridging the two is largely unidirectional: symbolic-to-audio controls drive a generator from a provided melody, chord, or drum track (Engel et al., 2020; Wu et al., 2022; 2024; Tsai et al., 2025; Tal et al., 2024; Lan et al., 2024; Lin et al., 2024; Novack et al., 2024), while transcription converts audio into the symbolic domain (Hawthorne et al., 2018; Gardner et al., 2022; Tan et al., 2024; Chang et al., 2024), with dense polyphony and timbrally similar instruments still being a bottleneck (Chaturvedi et al., 2026). U-MusT (Jung et al., 2026) jointly trains an encoder-decoder transformers, but relies on separate checkpoints rather than a single model. Across past work, one modality serves as conditioning and the other as output, where we have a shared generative process.

2.2 MUSIC SOURCE SEPARATION AND EDITING

Music source separation maps a mixture to its constituent stems, either as masked regression (Hennequin et al., 2020; Rouard et al., 2023; Lu et al., 2024) or by sampling from a generative prior over sources (Mariani et al., 2024; Chae & Lee, 2025; Zhu et al., 2022); works that bring in symbolic information (Hung et al., 2021; Lin et al., 2021) treat the score as a deterministic bottleneck or supervisory signal rather than directly modeling it.

Music editing changes targeted parts of a recording while leaving the rest intact, either at inference time by inverting the sampling trajectory (Manor & Michaeli, 2024), contrasting paired captions (Zhang et al., 2024), or steering toward the source (Niu et al., 2026) (no extra data needed, but inversion is approximate and drift is governed by hyperparameters rather than the edit itself), or at training time from (instruction, input, output) triplets (Wang et al., 2023; Zhang et al., 2025) or a conditioner on a frozen backbone (Tsai et al., 2024) (more reliable, but requires paired data per operation); in both cases, the edit is specified as text or a latent direction that never identifies which notes to change, so preservation is left implicit.

In all of the above, the symbolic representation enters as supervision or a fixed intermediate, never as a quantity the model learns to generate. MUSEDUO instead models audio and symbolic music jointly, and we show that coupling the two introduces various use cases.

3 METHOD

3.1 PRELIMINARIES

Diffusion transformers and rectified flow. Diffusion transformers (DiT) (Peebles & Xie, 2023) model diffusion over a sequence of latent tokens with a Transformer backbone. We use the rectified-flow variant (Liu et al., 2023b; Esser et al., 2024), which trains the DiT to predict, from a latent

$\mathbf{z}(t) = (1-t)\,\mathbf{z} + t\,\boldsymbol{\epsilon}$ noised along a linear path toward Gaussian noise ($\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, $t \in [0, 1]$), the velocity $\mathbf{v} = \mathbf{z} - \boldsymbol{\epsilon}$ that transports noise back to data; at inference, an ODE solver integrates the predicted velocity field from $t = 1$ to $t = 0$ to generate samples.

Multi-track generation. We build on Stemphonic (Wu et al., 2026b) to generate a flexible pack of *synchronized* audio tracks $\{\mathbf{z}_a^{(k)}\}_{k=1}^{K}$ in one pass as well as accompaniment generation, with variable track count K and instrument types. Treating each track as an independent batch element and assigning a *shared* initial noise $\boldsymbol{\epsilon}_a$ across the pack during training enforces synchronization. External audio and text conditions further support both from-scratch generation and accompaniment.

3.2 Overview

Our goal with MuseDuo is to output multi-modal musical tracks under flexible conditioning, where each track is a pair of time-aligned **audio** and **MIDI**: the audio is an isolated instrument recording, compressed by a pretrained VAE (Evans et al., 2026) into an audio latent $\mathbf{z}_a$ following latent diffusion practice, and the MIDI is similarly encoded into a latent $\mathbf{z}_m$, whose representation we detail in Section 3.4. We then use two types of user-specifiable conditions. *External* conditions are the reference audio $\mathcal{C}_{\text{audio}}$ and text instruction $\mathcal{C}_{\text{text}}$ (collectively $\mathcal{C}$): $\mathcal{C}_{\text{audio}}$ is the latent of a mix (or submix) of the same piece the target stem $\{\mathbf{z}_a, \mathbf{z}_m\}$ is from, and $\mathcal{C}_{\text{text}}$ specifies the task tag, target instrument, and optionally a music description (e.g., `a relaxed pop song ...`). *Internal* conditions are the modalities $\mathbf{z}_a, \mathbf{z}_m$ themselves: given unnoised, either can guide generation of the other, adding single-track **transcription** (audio→MIDI) and **rendering** (MIDI→audio).

3.3 Audio Conditioning for Reference-Based Tasks

Varying what fills the audio condition $\mathcal{C}_{\text{audio}}$ (Section 3.2) turns the same generative process into a family of reference-based tasks. Setting $\mathcal{C}_{\text{audio}}$ to a submix, as in Stemphonic (Section 3.1), yields **accompaniment**. We extend this to audio-domain **source separation** by instead setting, at training time, $\mathcal{C}_{\text{audio}}$ as the full audio mix, $\{\mathcal{C}_{\text{text}}^{(k)}\}_{k=1}^{K}$ as the individual instrument types we intend to extract, and $\{\mathbf{z}_a^{(k)}\}_{k=1}^{K}$ as the isolated instrument recordings (same for instrument removal). Paired with MuseDuo's joint audio-MIDI modeling, this setup also yields MIDI-domain **multi-track transcription** in the same inference pass. Restricting $\mathcal{C}_{\text{audio}}$ to a partial time span of the mix or a stem, rather than the whole track, further gives **audio inpainting** and other reference-conditioned tasks detailed in Section **??** and Table 1.

3.4 MIDI Variational Autoencoder

The MIDI sequence is a binary piano roll $\mathbf{m}$ (Dong et al., 2018) noting when each pitched note starts (*onset*), when it sounds / remains held (*frame*), and when each drum hit occurs (*onset* only); we consider 88 possible pitches (with onsets and frames) (Bittner et al., 2022) and 5 types of drum hits (Zehren et al., 2021), hence $\mathbf{m} \in \{0, 1\}^{(88+88+5)\times *}$. A naïve option to model this binary roll is treating it as-is (i.e., $\mathbf{z}_m := \mathbf{m}$). However, the distribution of $\mathbf{m}$ differs substantially from that of the Gaussian-priored $\mathbf{z}_a$, which may cause difficulties for joint modeling. We therefore propose MIDI variational autoencoder (MIDI-VAE), which reuses the 1D convolution + attention hybrid architecture of our audio VAE (Casebeer et al., 2026) and treats the 181 pitched note and drum rows as input (and output) channels to encode (and reconstruct), in the same way our audio VAE treats waveform channels. Since $\mathbf{m}$ has a different and relatively low time resolution (<100 Hz) compared to the audio waveform sampling rate (44.1 kHz), we adjust the decimation factors inside the VAE architecture such that the audio and MIDI latents share the same frame rate f, i.e., $\mathbf{z}_a \in \mathbb{R}^{\mathrm{C}_a \times \mathrm{T}f}$ and $\mathbf{z}_m \in \mathbb{R}^{\mathrm{C}_m \times \mathrm{T}f}$, where C_a, C_m are their respective latent channel dimensions, T is the duration (in seconds), and $\mathrm{T}f \in \mathbb{Z}$ is the latent sequence length (i.e., frame count). We train the MIDI-VAE following the standard KL-regularized reconstruction objective (Kingma & Welling, 2014), and can achieve near-perfect reconstruction (Table 11) with minimal hyperparameter tuning.

Table 1: **Inference configuration per task.** Each row gives the roles of the MIDI and audio latents, the two external conditions, and the distribution the inference pass samples from. 'target': denoised from $t = 1$ to $t = 0$; 'cond.': given clean at $t = 0$; †: produced in the same pass but not needed; $\star$: user edited. $\mathcal{C} := \{\mathcal{C}_{\text{audio}}, \mathcal{C}_{\text{text}}\}$, and $\mathcal{C}_{\text{audio}} = \mathbf{0}$ means text-only conditioning; 'p-span': a partial time span given. In $\mathcal{C}_{\text{text}}$, `<gmp>` is the global musical prompt, i.e., the free-form description of the music, and `<inst>` is the target instrument label. Separation and multi-track transcription are done in one pass with different fetched outputs. The composed tasks rely on two chained passes.

Task		$\mathbf{z}_a$	$\mathbf{z}_m$	$\mathcal{C}_{\text{audio}}$	$\mathcal{C}_{\text{text}}$	Target distribution
Text-conditioned						
Single-track (mix) generation		target	target†	$\mathbf{0}$	`generate, <gmp>, mix`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Multi-track (stem) generation		target	target†	$\mathbf{0}$	`generate, <gmp>, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Joint MIDI + audio generation		target	target	$\mathbf{0}$	`generate, <gmp>, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Text-to-MIDI		target†	target	$\mathbf{0}$	`generate, <gmp>, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Reference-conditioned						
Accompaniment		target	target†	submix	`accompany, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Separation / Removal		target	target†	mix	`separate / remove, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Multi-track transcription		target†	target	mix	`separate, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Audio inpainting (mix)		target	target†	mix p-span	`generate, <gmp>, mix`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Joint MIDI + audio inpainting (stem)		target	target	stem p-span	`generate, <gmp>, <inst>`	$q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$
Cross-modal						
Single-track transcription		cond.	target	$\mathbf{0}$	`generate, <inst>`	$q(\mathbf{z}_m \mid \mathbf{z}_a, \mathcal{C})$
MIDI-to-audio rendering		target	cond.	$\mathbf{0}$	`generate, <inst>`	$q(\mathbf{z}_a \mid \mathbf{z}_m, \mathcal{C})$
Composed (two passes)						
Timbre transfer	pass 1	cond.	target	$\mathbf{0}$	`generate, <(orig) inst>`	$q(\mathbf{z}_m \mid \mathbf{z}_a, \mathcal{C})$
	pass 2	target	cond.	$\mathbf{0}$	`generate, <(new) inst>`	$q(\mathbf{z}_a \mid \mathbf{z}_m, \mathcal{C})$
Note-level editing	pass 1	cond.	target	$\mathbf{0}$	`generate, <inst>`	$q(\mathbf{z}_m \mid \mathbf{z}_a, \mathcal{C})$
	pass 2	target	cond.$\star$	$\mathbf{z}_a$ p-span (from pass 1)	`generate, <inst>`	$q(\mathbf{z}_a \mid \mathbf{z}_m, \mathcal{C})$

3.5 Unified Audio-MIDI DiT

Our DiT backbone jointly denoises $\{\mathbf{z}_a, \mathbf{z}_m\}$, applying the rectified-flow noising of Section 3.1 to each modality: $\mathbf{z}_a(t_a)$ and $\mathbf{z}_m(t_m)$, with separate per-modality timesteps $t_a, t_m \in [0, 1]$ and noises $\boldsymbol{\epsilon}_a, \boldsymbol{\epsilon}_m \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We then focus on two critical aspects of multi-modal generation below.

Decoupled noise timesteps. The coupling of the timestep (t_a, t_m) during training is a key design decision: a shared timestep exposes the model only to equally-noised (audio, MIDI) pairs, whereas decoupled timesteps expose it to all combinations of noise levels. We adopt a hybrid design: for 50% of training samples, we draw t_a and t_m independently from the uniform distribution $\mathcal{U}(0, 1)$, and for the remaining 50%, we use a coupled noise level $t_a = t_m := t$, where $t \sim \mathcal{U}(0, 1)$. At inference, letting $q(\cdot \mid \cdot)$ denote the model's learned conditional distribution, (t_a, t_m) can likewise be set separately, with each choice selecting a different conditional over $\{\mathbf{z}_a, \mathbf{z}_m\}$ to sample from: setting $t_a = t_m$ and denoising both latents from $t = 1$ to 0 gives joint generation $q(\mathbf{z}_a, \mathbf{z}_m \mid \mathcal{C})$, while fixing one timestep at 0 and denoising only the other gives the cross-modal transcription and rendering conditionals $q(\mathbf{z}_m \mid \mathbf{z}_a, \mathcal{C})$ and $q(\mathbf{z}_a \mid \mathbf{z}_m, \mathcal{C})$ (full task-by-task mapping in Table 1). Crucially, if the model were only trained on coupled $t_a = t_m$ timesteps, these two cross-modal conditionals would both be out-of-domain at inference.

Input fusion. Both noised latents first pass through their own input projections — the audio patch embedder ϕ_a and roll embedder ϕ_m — projecting the C_a- and C_m-channel sequences frame-wise to the shared DiT hidden dimension d, yielding $\mathbf{h}_a = \phi_a(\mathbf{z}_a(t_a))$ and $\mathbf{h}_m = \phi_m(\mathbf{z}_m(t_m))$, both $\in \mathbb{R}^{d \times \mathrm{T}f}$. We consider two ways of fusing them,

$$\mathbf{h}_{\text{add}} = \mathbf{h}_a + \mathbf{h}_m \in \mathbb{R}^{d \times \mathrm{T}f}, \qquad \text{or} \qquad \mathbf{h}_{\text{cat}} = [\mathbf{h}_a,\, \mathbf{h}_m] \in \mathbb{R}^{d \times 2\mathrm{T}f}, \tag{1}$$

where $[\,\cdot\,,\,\cdot\,]$ denotes concatenation along the sequence-length axis. We adopt *sequence concatenation* as our default, which keeps each modality's own hidden-state positions and lets cross-modal information exchange happen through self-attention, at the cost of a doubled sequence length; we encourage learning the cross-domain frame correspondence it requires via shared RoPE (Su et al., 2024) indices for time-aligned (MIDI, audio) frame pairs. Trade-off discussion in Appendix C.

Output heads and objective. A single DiT processes the fused feature sequence to produce the output representation $\tilde{\mathbf{h}} = \mathrm{DiT}_\theta(\mathbf{h}_a, \mathbf{h}_m;\, t_a, t_m, \mathcal{C})$, where θ denotes the DiT's parameters. Two

modality-wise output heads ψ_a and ψ_m then map $\tilde{\mathbf{h}}$ back to the C_a- and C_m-channel latent spaces. Under addition fusing, both heads read the same $\mathrm{T}f$ output hidden states; under sequence concatenation, $\tilde{\mathbf{h}} = [\tilde{\mathbf{h}}_a, \tilde{\mathbf{h}}_m]$ is split back into its audio and MIDI halves and each head reads its own half. In both cases, each head predicts the rectified-flow velocity of its corresponding stream, i.e., $\hat{\mathbf{v}}_a \in \mathbb{R}^{\mathrm{C}_a \times \mathrm{T}f}$ and $\hat{\mathbf{v}}_m \in \mathbb{R}^{\mathrm{C}_m \times \mathrm{T}f}$, in one pass. The two predictions are concatenated along the channel axis and regressed onto the ground-truth velocities with a single flow objective,

$$\mathcal{L} = \mathbb{E} \left\| [\hat{\mathbf{v}}_a;\, \hat{\mathbf{v}}_m] - [\mathbf{z}_a - \boldsymbol{\epsilon}_a;\, \mathbf{z}_m - \boldsymbol{\epsilon}_m] \right\|_2^2, \tag{2}$$

where $[\,\cdot\,;\,\cdot\,]$ means channel-wise concatenation and the expectation is taken over $(\mathbf{z}_a, \mathbf{z}_m, t_a, t_m, \mathcal{C})$.

3.6 Multi-task Data Creation

The vast majority of real-world music has no paired MIDI: existing audio-MIDI corpora are either small, genre-narrow, or synthetically rendered (e.g., Slakh), which would otherwise confine a joint audio-MIDI model to a small fraction of the musical diversity available in audio alone. We sidestep this bottleneck by extracting per-stem MIDI on the fly from ordinary stems using frozen transcribers, letting MuseDuo train on any licensed multi-track recording rather than the narrow slice of music with existing MIDI annotations, as shown in Fig. 2. Summing a song's stems gives the mix; blending mix, stems, and extracted MIDI in different combinations for our input-output pair training data.

4 Experiments

MuseDuo supports all applications in Table 1 by setting the roles of $(\mathbf{z}_a, \mathbf{z}_m, \mathcal{C})$ at inference; *mix* generation and inpainting are retained from our pretrained backbone. We evaluate a single checkpoint on 10 tasks: mix and multi-track generation, separation, single- and multi-track transcription, timbre transfer (TT), MIDI-to-audio rendering (M2A), text-to-MIDI generation, and note-level editing. Training and architectural details are in Appendix A.

4.1 Metrics

For **audio quality** we use FAD (Kilgour et al., 2019) and KAD (Chung et al., 2025) between generated and reference audio, computed on MusicFM (Won et al., 2024) and VGGish (Hershey et al., 2017) embeddings (subscripts M and V). For **separation quality**, we use SDR improvement over the mixture (SDRi) (Vincent et al., 2006) and the L2 distance between mel spectrograms of the output and the target (Mel-L2), which ignores phase and thus does not penalize generated stems that sound correct but are not waveform-aligned. For **symbolic alignment**, we use note-level Onset, Frame, and Drums F1 (Hawthorne et al., 2018) against reference MIDI. For **text and audio adherence**, we use CLAP score (Wu et al., 2023) for generation and target-instrument timbre in MIDI-to-Audio, note-level editing, and editing. For text-to-MIDI, we compute CLAP on General MIDI renderings and report Chroma and Groove Jensen–Shannon distances (JSD) to reference MIDI.

4.2 Baselines

As no prior model covers all 10 tasks, we compare against specialist baselines per task. **Mix generation:** MusicGen–M (Copet et al., 2023), Stable Audio 3–M (Evans et al., 2026), Stemphonic (Wu et al., 2026b)* extended with mix generation, and our pretrained base model. **Multi-track generation:** Stemphonic, and MGE-LDM (Chae & Lee, 2025). **Separation:** open-vocabulary methods only, namely MGE-LDM and Stemphonic* extended with separation (Section 3.3); the latter shares our training data but models audio only, isolating the effect of joint MIDI modeling. **Transcription:** Basic Pitch (Bittner et al., 2022) and ADTOF (Zehren et al., 2021) for single track transcription baseline, and MuScriptor (Rouard et al., 2026) (S/M/L: 0.1B/0.3B/1.3B) for both single and multi-track baseline. **MIDI-to-Audio and Timbre transfer:** Spectrogram Diffusion (Hawthorne et al., 2022), MuseControlLite (Tsai et al., 2025) using only melody condition, and MIDI-DDSP (Wu et al., 2022). **Note editing:** MuseControlLite conditioned on the both audio inpainting and melody condition. **Text-to-MIDI:** Text2midi (Bhandari et al., 2025) and MIDI-LLM (Wu et al., 2026a).

4.3 Evaluation Datasets

MuseDuo is trained only on licensed internal data, and all evaluation sets are unseen. **Mix generation:** Song Describer (Manco et al., 2023), a standard TTM benchmark whose human-written captions test prompt adherence. **Multi-track generation and separation:** MoisesDB (Pereira et al., 2023) and MUSDB18 (Rafii et al., 2017), totaling 390 real songs with isolated stems. MUSDB18 is the standard separation benchmark, and the fine-grained stem taxonomy of MoisesDB tests open-vocabulary instrument control. **Transcription, MIDI-to-Audio and Timbre transfer:** Slakh (Manilow et al., 2019), pop/rock synthesized from Lakh MIDI (Raffel, 2016), and URMP (Montesinos et al., 2020), 44 real classical ensemble pieces, both with aligned stem-level MIDI. The two are complementary: Slakh offers dense multi-instrument arrangements with exact MIDI, while URMP tests generalization to real acoustic performances. **Note editing:** URMP is used, since we want to evaluate if the edit reserves the timbre of real instrument after editing. **Text-to-MIDI:** MidiCaps (Melechovsky et al., 2024) captions with paired Lakh MIDI (Raffel, 2016).

4.4 Single-track Generation

We evaluate standard single-track TTM generation. Both our base model and the fine-tuned MuseDuo outperform other baselines on audio realism, though both fall slightly behind Stable Audio–M (Evans et al., 2026) in text-audio adherence. Comparing our model before and after fine-tuning, there is only a marginal impact on mix quality.

Table 2: **Single-track TTM mix generation** results on Song Describer. (*: our improved re-implementation.)

	Quality				Text Ctrl.
Model	$KAD_M \downarrow$	$FAD_M \downarrow$	$KAD_V \downarrow$	$FAD_V \downarrow$	CLAP↑
MusicGen–M	8.05	35.9	4.29	2.00	0.309
Stable Audio–M	2.08	15.7	3.10	1.60	**0.343**
Stemphonic*	2.86	17.1	4.01	1.79	0.312
Ours (base)	**1.93**	**12.5**	2.64	1.29	0.326
Ours	2.31	13.6	**2.38**	**1.07**	0.325

4.5 Multi-track Generation

We evaluate multi-track TTM generation, where the model generates a variable set of synchronized instrument stems from a text prompt. We use Qwen3-omni (Xu et al., 2025) to caption the evaluation datasets, and use it as text conditions. As shown in Table 3, MuseDuo outperforms Stemphonic on all metrics, showing that adding the MIDI modality and additional tasks does not come at the cost of multi-track audio generation. MuseDuo also outperforms MGE-LDM by a large margin across all metrics, despite MGE-LDM being trained in-domain on these datasets.

Table 3: **Multi-track generation** results on MoisesDB + MusDB.

		Quality		Text Ctrl.
Test set	Model	$KAD_M \downarrow$	$FAD_V \downarrow$	CLAP↑
Full (2.5K samples)	Stemphonic	5.98	1.28	0.277
	Ours	**4.26**	**0.72**	**0.316**
MGE-LDM held-out (0.5K samples)	MGE-LDM	25.35	1.50	0.156
	Ours	**4.91**	**0.85**	**0.276**

Table 4: **Source separation** results on MoisesDB + MusDB. (*: our improved re-implementation.)

		Correctness		Quality	
Test set	Model	SDRi↑	Mel-L2↓	$KAD_M \downarrow$	$FAD_V \downarrow$
Full (10K samples)	Stemphonic*	3.71	3.68	1.00	0.33
	Ours	**5.85**	**2.61**	**0.87**	**0.32**
MGE-LDM held-out (2K samples)	MGE-LDM	1.42	3.74	5.93	0.50
	Ours	**5.92**	**2.49**	**1.07**	**0.43**

4.6 Source Separation

As shown in Table 4, MuseDuo outperforms Stemphonic* on all metrics, with notable gains in separation correctness. MuseDuo also outperforms MGE-LDM by a large margin across all metrics. Moreover, unlike both baselines, MuseDuo produces the MIDI transcription of each separated stem in the same inference pass, naturally enabling multi-track transcription (Section 4.7).

4.7 Transcription

For multi-track transcription, where each instrument must be transcribed directly from the mixture, MuseDuo outperforms all MuScriptor variants on every metric and dataset (Table 5, left), with large gains in Frame F1. We attribute this to performing separation and transcription jointly in a

Table 5: **Multi-** and **single-track transcription** results on Slakh (*pop / rock*, left & black) and URMP (*classical*, right & violet).

Model		Multi-track Onset F1 ↑	Multi-track Frame F1 ↑	Multi-track Drums F1 ↑	Single-track Onset F1 ↑	Single-track Frame F1 ↑	Single-track Drums F1 ↑
Basic Pitch		– single-track only –			0.527 · 0.826	0.653 · **0.906**	—
ADTOF		– single-track only –			—	—	0.757
MuScriptor	S	0.232 · 0.403	0.256 · 0.463	0.649	0.604 · 0.831	0.622 · 0.871	0.760
	M	0.266 · 0.448	0.281 · 0.523	0.688	0.679 · **0.850**	0.663 · 0.879	**0.789**
	L	0.305 · 0.443	0.316 · 0.537	0.686	**0.697** · 0.780	**0.685** · 0.890	0.779
Ours		**0.316** · **0.506**	**0.449** · **0.613**	**0.707**	0.508 · 0.813	0.654 · 0.904	0.742

Table 6: **MIDI-to-audio** (**M2A**) and **timbre transfer** (**TT**) results on Slakh (*pop / rock*, left & black) and URMP (*classical*, right & violet). (‡ Spectrogram Diff. uses MuScriptor-medium transcription front-end on TT; and, is not evaluated on URMP for using it as training (trn) data.)

	MIDI-to-Audio (M2A)				Timbre Transfer (TT)			
	Note Ctrl.		Timbre	Quality	Note Ctrl.		Timbre	Quality
Model	Onset F1 ↑	Frame F1 ↑	CLAP ↑	KAD ↓	Onset F1 ↑	Frame F1 ↑	CLAP ↑	KAD ↓
Spectrogram Diff.‡	0.338 · trn	0.520 · trn	0.195 · trn	23.2 · trn	0.157 · trn	0.371 · trn	0.178 · trn	25.5 · trn
MuseControlLite	0.544 · 0.573	0.640 · 0.730	0.239 · 0.282	7.2 · **22.8**	0.403 · 0.745	0.489 · 0.848	0.176 · 0.290	6.0 · **16.9**
Ours	**0.794** · **0.844**	**0.857** · **0.893**	**0.243** · **0.349**	**3.8** · 23.9	**0.520** · **0.752**	**0.607** · **0.853**	**0.237** · **0.321**	**3.2** · 28.1

single pass, which lets each stem's MIDI benefit from the model's open-vocabulary separation ability. For single-track transcription (Table 5, right), MUSEDUO performs on par with Basic Pitch and ADTOF, which serve as the teachers for our on-the-fly MIDI extraction during training, suggesting that its single-track accuracy is largely bounded by the teacher. MuScriptor is higher Onset F1 on Slakh, while MUSEDUO is competitive on URMP, outperforming all MuScriptor sizes in Frame F1.

4.8 MIDI-TO-AUDIO RENDERING, TIMBRE TRANSFER

As shown in Table 6, we outperform all baselines in note control and timbre adherence on both datasets, by a large margin for MIDI-to-audio. Unlike the melody-conditioned MuseControlLite, our MIDI latent specifies onsets and polyphony on the audio's own time grid. For timbre transfer, we still lead even though the baselines receive stronger MuScriptor-M transcription input, while MUSEDUO uses its own. The one exception is audio quality on URMP, where MuseControlLite scores better on KAD. Since MIDI-DDSP is trained on URMP and supports only its 13 instruments, we evaluate it on the URMP test split and on a held-out Slakh subset restricted to those instruments (Table 7). MUSEDUO achieves better note control on both tasks and comparable timbre.

4.9 NOTE-LEVEL EDITING

We test two note edit types: transposing a span by a third, fourth, or fifth, and major and minor swap. In Table 8, MUSEDUO preserves the non-edited spans better than MuseControlLite, while also achieving higher timbre adherence and quality. For span transposition, it also follows the target more closely. For major↔minor, MuseControlLite achieves higher editability, but with worse fidelity, timbre, and quality, suggesting it follows the target at the cost of the original performance. This edit alters only a few scale degrees, so it tests fine-grained note control more strictly than transposition.

4.10 TEXT-TO-MIDI GENERATION AND JOINT AUDIO-MIDI CONSISTENCY

Finally, we evaluate multi-track text-to-MIDI generation on MidiCaps, scoring the MIDI output of joint generation (Table 1). In Table 9, MUSEDUO achieves the highest CLAP score despite being trained without ground-truth MIDI, while both symbolic baselines obtain lower Chroma and Groove JSD. However, the baselines are trained on MidiCaps, matching the reference distribution, whereas MUSEDUO learns MIDI from transcriptions of audio stems with performance-like timing that differs from hand-authored MIDI.

Table 7: **MIDI-to-audio / timbre transfer** vs. MIDI-DDSP on curated monophonic Slakh samples of its *in-domain* instruments. (n=31 / 77 for **M2A / TT**; ‡MIDI-DDSP uses MuScriptor-medium for **TT**.)

Model	Onset F1 ↑	Frame F1 ↑	CLAP ↑
MIDI-DDSP‡	0.744 · 0.329	0.785 · 0.443	0.311 · **0.301**
Ours	**0.808** · **0.374**	**0.799** · **0.535**	**0.328** · 0.288

Table 8: **Note-level editing** results on URMP on two example tasks: **Span Transposition** of ±3rd, ±4th, ±5th diatonic intervals (left) and **Major↔Minor** key change (right) .

Model	**Fidelity** Mel-L2 ↓	**Editability** Onset F1 ↑	**Timbre** CLAP ↑	**Quality** KAD ↓
MuseControlLite	0.818 · 0.882	0.324 · **0.504**	0.261 · 0.242	34.0 · 18.6
Ours	**0.167** · **0.305**	**0.658** · 0.423	**0.386** · **0.376**	**2.1** · **2.8**

Table 10: **Ablations of MUSEDUO** across tasks (rows 2–6 each alter one attribute from the default, top row; Section 4.11). Noise = per-modality noise timestep t schedule ($\mathcal{U}$ uniform, $\mathcal{LN}$ logit-normal); $t_a \perp t_m$ = fraction of steps with decoupled audio-MIDI noising timesteps (0.5 = half examples decoupled, 0.0 = always tied). Task and evaluation datasets shown here are: **Multi-track generation / Separation** on MoisesDB+MusDB; **Transcription/M2A/TT** on Slakh; **MIDI generation** on Lakh. Severe degradation highlighted in red.

Config					**MT Generation**		**Sep.**	**ST Trans.**	**MT Trans.**	**M2A**	**TT**	**MIDI Gen.**
MIDI-VAE	$t_a \perp t_m$	Noise	Fuse	PE-align	KAD_M ↓	CLAP ↑	SDRi ↑	Onset ↑	Onset ↑	Onset ↑	Onset ↑	CLAP ↑
✓	0.5	$\mathcal{U}$	cat	✓	**4.26**	0.316	5.85	**0.508**	**0.316**	**0.794**	**0.520**	0.321
✓	0.5	$\mathcal{U}$	cat	✗	4.43	0.314	5.77	0.506	0.315	0.771	0.506	0.320
✓	0.5	$\mathcal{U}$	add	—	4.59	**0.320**	4.99	**0.508**	0.299	0.762	0.495	**0.324**
✓	0.5	$\mathcal{LN}$	cat	✓	4.72	0.306	2.46	0.446	0.219	0.787	0.508	0.317
✓	0.0	$\mathcal{U}$	cat	✓	4.44	0.312	5.78	0.006	0.274	0.008	0.004	**0.324**
✗	0.5	$\mathcal{U}$	cat	✓	4.70	0.308	**6.18**	0.047	0.040	0.742	0.027	0.106

Also, unlike baselines, MUSEDUO does not require an external synthesizer as it jointly produces audio outputs. We measure the joint audio-MIDI consistency by transcribing output audio back to MIDI, resulting in (Onset, Frame, Drums) F1 scores of (0.61, 0.82, 0.63), roughly in line with the performance of Basic Pitch & ADTOF transcribers themselves (Table 5, single-track), which we rely on for MIDI data extraction.

Table 9: **Multi-track text-to-MIDI generation** results on MidiCaps / Lakh MIDI.

	Text Ctrl.	**Musical Quality**	
Model	CLAP ↑	Chroma JS ↓	Groove JS ↓
Text2midi	0.303	0.135	0.291
MIDI-LLM	0.316	**0.093**	**0.236**
Ours	**0.321**	0.168	0.417

4.11 ABLATION STUDY

Table 10 alters one design choice of the default configuration (top row) at a time; the default ranks first or second on every metric. **Positional alignment:** removing shared RoPE indices for audio and MIDI causes small drops on all metrics, largest on MIDI-to-audio and timbre transfer. **Fusion:** addition matches concatenation on text adherence but degrades separation, MIDI-to-audio, and timbre transfer, where the audio must closely follow a reference. We note that concatenation incurs slightly slower inference: On an A100 GPU, it takes 3.4 / 7.3 seconds end-to-end for 32-sec. 1-/4-track outputs, vs. 3.2 / 4.5 s for addition; both variants are, nevertheless, much faster than real time. **Noise schedule:** the logit-normal schedule of our base model rarely samples extreme noise levels and severely degrades separation and multi-track transcription. **Decoupled timesteps:** always tying $t_a = t_m$ keeps joint generation largely intact but collapses all cross-modal tasks to near-zero Onset F1, as their conditionals are never seen in training (Section 3.5). **MIDI-VAE:** modeling the raw piano roll $\mathbf{m}$ collapses every task that generates MIDI (transcription, text-to-MIDI, and timbre transfer), while MIDI-to-audio, which only reads the roll $\mathbf{m}$ as a condition still works.

5 CONCLUSION

We present MUSEDUO, a unified diffusion/flow model for audio-MIDI generation, rendering, transcription, separation, and editing, built by fine-tuning a pretrained TTM backbone with a new MIDI-VAE, on-the-fly MIDI extraction, and a concise multi-task training recipe. Across 10 tasks and against specialist baselines, MUSEDUO is comparable or better on almost all tasks and metrics, enabling a wide range of creative music applications.

ETHICS STATEMENT

We recognize that music generation can be a sensitive topic. Our focus is on advancing human creative control in music generation and building tools that are genuinely useful to musicians. We train on licensed, internal data, and evaluate on a variety of open-source, Creative Commons datasets.

AI USE DISCLOSURE

In this work, we used generative AI tools for coding assistance and writing assistance: LLM-based assistants helped write and debug parts of our training and evaluation code, and helped draft and revise parts of the manuscript text. We have not used generative AI tools for research ideation, experimental design, or the production or analysis of experimental results, but we used them for the discovery and categorization of related work. Additionally, we used generative AI tools for language polishing and LaTeX formatting (e.g., table layout). We have reviewed all AI-assisted work: AI-assisted code was reviewed by the authors and validated by running it and checking its outputs, and AI-assisted text was revised by the authors, with every numerical result, claim, and citation checked against our experimental results and the cited sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

In our work, we focus on developing clear, reproducible scientific insights by comparing our approach against a wide variety of baseline methods, including both open-source implementations and our own re-implementations, on standard evaluation metrics. All evaluation datasets (Song Describer, Slakh, URMP, MoisesDB, MusDB, MidiCaps/Lakh MIDI) are public and standard, so comparisons against baselines are independently verifiable even though MUSEDUO itself is trained on licensed, proprietary data; given the complexities of music licensing, we do not plan to open-source our training code, data pipeline, or model checkpoints. One evaluation setting carries a specific dependency worth noting explicitly: single-track transcription is evaluated against the same class of model (Basic Pitch, ADTOF) used as the on-the-fly teacher for our training targets, so its absolute performance is bounded by teacher quality. Multi-track transcription, separation, and generation results carry no such dependency and are the stronger evidence of the benefits of joint audio-MIDI modeling.

REFERENCES

Andrea Agostinelli, Timo I Denk, Zalán Borsos, Jesse Engel, Mauro Verzetti, Antoine Caillon, Qingqing Huang, Aren Jansen, Adam Roberts, Marco Tagliasacchi, et al. MusicLM: Generating music from text. *arXiv preprint arXiv:2301.11325*, 2023.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *International Conference on Machine Learning (ICML)*. PMLR, 2023.

Keshav Bhandari, Abhinaba Roy, Kyra Wang, Geeta Puri, Simon Colton, and Dorien Herremans. Text2midi: Generating symbolic music from captions. In *AAAI Conference on Artificial Intelligence*, 2025.

Rachel M Bittner, Juan José Bosch, David Rubinstein, Gabriel Meseguer-Brocal, and Sebastian Ewert. A lightweight instrument-agnostic model for polyphonic note transcription and multipitch estimation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2022.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. AudioLM: A language modeling approach to audio generation. *Transactions on Audio, Speech, and Language Processing (TASLP)*, 2023.

Jonah Casebeer, Ge Zhu, Zhepei Wang, and Nicholas J Bryan. A generative-first neural audio autoencoder. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. IEEE, 2026.

Yunkee Chae and Kyogu Lee. MGE-LDM: Joint latent diffusion for simultaneous music generation and source extraction. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Sungkyun Chang, Emmanouil Benetos, Holger Kirchhoff, and Simon Dixon. YourMT3+: Multi-instrument music transcription with enhanced transformer architectures and cross-dataset stem augmentation. In *International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, 2024.

Ojas Chaturvedi, Kayshav Bhardwaj, Tanay Gondil, Benjamin Shiue-Hal Chou, Kristen Yeon-Ji Yun, Yung-Hsiang Lu, Yujia Yan, and Sungkyun Chang. Advancing multi-instrument music transcription: Results from the 2025 AMT challenge. *arXiv preprint arXiv:2603.27528*, 2026.

Ke Chen, Yusong Wu, Haohe Liu, Marianna Nezhurina, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. MusicLDM: Enhancing novelty in text-to-music generation using beat-synchronous mixup strategies. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J Weiss, Mohammad Norouzi, and William Chan. WaveGrad: Estimating gradients for waveform generation. In *International Conference on Learning Representations (ICLR)*, 2021.

Yoonjin Chung, Pilsun Eu, Junwon Lee, Keunwoo Choi, Juhan Nam, and Ben Sangbae Chon. KAD: No more FAD! An effective and efficient evaluation metric for audio generation. *arXiv preprint arXiv:2502.15602*, 2025.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang, and Yi-Hsuan Yang. MuseGAN: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment. In *AAAI Conference on Artificial Intelligence*, 2018.

Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. GANSynth: Adversarial neural audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2019.

Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. DDSP: Differentiable digital signal processing. In *International Conference on Learning Representations (ICLR)*, 2020.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *International Conference on Machine Learning (ICML)*, 2024.

Zach Evans, CJ Carr, Josiah Taylor, Scott H Hawley, and Jordi Pons. Fast timing-conditioned latent audio diffusion. In *International Conference on Machine Learning (ICML)*, 2024.

Zach Evans, Julian D Parker, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio open. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025.

Zach Evans, Julian D Parker, Matthew Rice, CJ Carr, Zack Zukowski, Josiah Taylor, and Jordi Pons. Stable audio 3. *arXiv preprint arXiv:2605.17991*, 2026.

Josh Gardner, Ian Simon, Ethan Manilow, Curtis Hawthorne, and Jesse Engel. MT3: Multi-task multitrack music transcription. In *International Conference on Learning Representations (ICLR)*, 2022.

Junmin Gong, Yulin Song, Wenxiao Zhao, Sen Wang, Shengyuan Xu, Jing Guo, and Xuerui Yang. ACE-Step 1.5: Pushing the boundaries of open-source music generation. *arXiv preprint arXiv:2602.00744*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Curtis Hawthorne, Erich Elsen, Jialin Song, Adam Roberts, Ian Simon, Colin Raffel, Jesse Engel, Sageev Oore, and Douglas Eck. Onsets and frames: Dual-objective piano transcription. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2018.

Curtis Hawthorne, Ian Simon, Adam Roberts, Neil Zeghidour, Josh Gardner, Ethan Manilow, and Jesse Engel. Multi-instrument music synthesis with spectrogram diffusion. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2022.

Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation tool with pre-trained models. *Journal of Open Source Software (JOSS)*, 2020.

Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. CNN architectures for large-scale audio classification. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017.

Cheng-Zhi Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M Dai, Matthew D Hoffman, Monica Dinculescu, and Douglas Eck. Music transformer. In *International Conference on Learning Representations (ICLR)*, 2019.

Yu-Siang Huang and Yi-Hsuan Yang. Pop music transformer: Beat-based modeling and generation of expressive pop piano compositions. In *ACM Multimedia*, 2020.

Yun-Ning Hung, Gordon Wichern, and Jonathan Le Roux. Transcription is all you need: Learning to separate musical mixtures with score as supervision. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.

Jongmin Jung, Dongmin Kim, Sihun Lee, Seola Cho, Hyungjoon Soh, Irmak Bukey, Chris Donahue, and Dasaem Jeong. U-MusT: A unified framework for cross-modal translation of score images, symbolic music, and performance audio. *Transactions on Audio, Speech and Language Processing (TASLP)*, 34:1876–1891, 2026. doi: 10.1109/TASLPRO.2025.3648794.

Tornike Karchkhadze, Mohammad Rasool Izadi, Ke Chen, Gerard Assayag, and Shlomo Dubnov. Multi-track MusicLDM: Towards versatile music generation with latent diffusion model. In *International Conference on ArtsIT, Interactivity and Game Creation*. Springer, 2024.

Tornike Karchkhadze, Mohammad Rasool Izadi, and Shlomo Dubnov. Simultaneous music separation and generation using multi-track latent diffusion models. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025.

Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms. In *Interspeech*, 2019.

Gwanghyun Kim, Alonso Martinez, Yu-Chuan Su, Brendan Jou, José Lezama, Agrim Gupta, Lijun Yu, Lu Jiang, Aren Jansen, Jacob Walker, et al. A versatile diffusion transformer with mixture of noise levels for audiovisual generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations (ICLR)*, 2021.

Yun-Han Lan, Wen-Yi Hsiao, Hao-Chung Cheng, and Yi-Hsuan Yang. MusiConGen: Rhythm and chord control for transformer-based text-to-music generation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

Matthew Le, Apoorv Vyas, Bowen Shi, Brian Karrer, Leda Sari, Rashel Moritz, Mary Williamson, Vimal Manohar, Yossi Adi, Jay Mahadeokar, et al. Voicebox: Text-guided multilingual universal speech generation at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Wei-Jaw Lee, Fang-Chih Hsieh, Xuanjun Chen, Fang-Duo Tsai, and Yi-Hsuan Yang. Training-efficient text-to-music generation with state-space modeling. *arXiv preprint arXiv:2601.14786*, 2026.

Liwei Lin, Qiuqiang Kong, Junyan Jiang, and Gus Xia. A unified model for zero-shot music source separation, transcription and synthesis. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2021.

Liwei Lin, Gus Xia, Junyan Jiang, and Yixiao Zhang. Content-based controls for music large language modeling. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

Haohe Liu, Zehua Chen, Yi Yuan, Xinhao Mei, Xubo Liu, Danilo Mandic, Wenwu Wang, and Mark D Plumbley. AudioLDM: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning (ICML)*, 2023a.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations (ICLR)*, 2023b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Peiling Lu, Xin Xu, Chenfei Kang, Botao Yu, Chengyi Xing, Xu Tan, and Jiang Bian. MuseCoco: Generating symbolic music from text. *arXiv preprint arXiv:2306.00110*, 2023.

Wei-Tsung Lu, Ju-Chiang Wang, Qiuqiang Kong, and Yun-Ning Hung. Music source separation with band-split RoPE transformer. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

Ilaria Manco, Benno Weck, Seungheon Doh, Minz Won, Yixiao Zhang, Dmitry Bogdanov, Yusong Wu, Ke Chen, Philip Tovstogan, Emmanouil Benetos, et al. The song describer dataset: a corpus of audio captions for music-and-language evaluation. In *NeurIPS Workshop on Machine Learning for Audio*, 2023.

Ethan Manilow, Gordon Wichern, Prem Seetharaman, and Jonathan Le Roux. Cutting music source separation some slakh: A dataset to study the impact of training data quality and quantity. In *Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. IEEE, 2019.

Hila Manor and Tomer Michaeli. Zero-shot unsupervised and text-based audio editing using DDPM inversion. In *International Conference on Machine Learning (ICML)*, 2024.

Giorgio Mariani, Irene Tallini, Emilian Postolache, Michele Mancusi, Luca Cosmo, and Emanuele Rodolà. Multi-source diffusion models for simultaneous music generation and separation. In *International Conference on Learning Representations (ICLR)*, 2024.

Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio. SampleRNN: An unconditional end-to-end neural audio generation model. In *International Conference on Learning Representations (ICLR)*, 2017.

Jan Melechovsky, Abhinaba Roy, and Dorien Herremans. MidiCaps: A large-scale MIDI dataset with text captions. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

Lejun Min, Junyan Jiang, Gus Xia, and Jingwei Zhao. Polyffusion: A diffusion model for polyphonic score generation with internal and external controls. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2023.

Juan F Montesinos, Olga Slizovskaia, and Gloria Haro. Solos: A dataset for audio-visual music analysis. In *International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020.

Xinlei Niu, Kin Wai Cheuk, Jing Zhang, Naoki Murata, Chieh-Hsin Lai, Michele Mancusi, Woosung Choi, Giorgio Fabbro, Wei-Hsiang Liao, Charles Patrick Martin, et al. SteerMusic: Enhanced musical consistency for zero-shot text-guided and personalized music editing. In *AAAI Conference on Artificial Intelligence*, 2026.

Zachary Novack, Julian McAuley, Taylor Berg-Kirkpatrick, and Nicholas J Bryan. DITTO: Diffusion inference-time T-optimization for music generation. In *International Conference on Machine Learning (ICML)*, 2024.

Julian D Parker, Janne Spijkervet, Katerina Kosta, Furkan Yesiler, Boris Kuznetsov, Ju-Chiang Wang, Matt Avent, Jitong Chen, and Duc Le. StemGen: A music generation model that listens. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *International Conference on Computer Vision (ICCV)*, 2023.

Igor Pereira, Felipe Araújo, Filip Korzeniowski, and Richard Vogl. MoisesDB: A dataset for source separation beyond 4-stems. In *International Society for Music Information Retrieval Conference (ISMIR)*, Milan, Italy, 2023.

Emilian Postolache, Giorgio Mariani, Luca Cosmo, Emmanouil Benetos, and Emanuele Rodolà. Generalized multi-source inference for text conditioned music diffusion models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

Xingwei Qu, Yuelin Bai, Yinghao Ma, Ziya Zhou, Ka Man Lo, Jiaheng Liu, Ruibin Yuan, Lejun Min, Xueling Liu, Tianyu Zhang, Xinrun Du, et al. MuPT: A generative symbolic music pretrained transformer. In *International Conference on Learning Representations (ICLR)*, 2025.

Colin Raffel. *Learning-based methods for comparing sequences, with applications to audio-to-MIDI alignment and matching*. PhD thesis, Columbia University, 2016.

Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017. URL https://doi.org/10.5281/zenodo.1117372.

Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid transformers for music source separation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

Simon Rouard, Robin San Roman, Yossi Adi, and Axel Roebel. MusicGen-Stem: Multi-stem music generation and edition through autoregressive modeling. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025.

Simon Rouard, Michael Krause, Axel Roebel, Carl-Johann Simon-Gabriel, and Alexandre Défossez. MuScriptor: An open model for multi-instrument music transcription. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2026.

Ludan Ruan, Yiyang Ma, Huan Yang, Huiguo He, Bei Liu, Jianlong Fu, Nicholas Jing Yuan, Qin Jin, and Baining Guo. MM-Diffusion: Learning multi-modal diffusion models for joint audio and video generation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2023.

Flavio Schneider, Ojasv Kamal, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Efficient text-to-music diffusion models. In *Association for Computational Linguistics (Volume 1: Long Papers)*, 2024.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Or Tal, Alon Ziv, Itai Gat, Felix Kreuk, and Yossi Adi. Joint audio and symbolic conditioning for temporally controlled text-to-music generation. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

Hao Hao Tan, Kin Wai Cheuk, Taemin Cho, Wei-Hsiang Liao, and Yuki Mitsufuji. MR-MT3: Memory retaining multi-track music transcription to mitigate instrument leakage. *arXiv preprint arXiv:2403.10024*, 2024.

John Thickstun, David Hall, Chris Donahue, and Percy Liang. Anticipatory music transformer. *Transactions on Machine Learning Research (TMLR)*, 2024.

Fang-Duo Tsai, Shih-Lun Wu, Haven Kim, Bo-Yu Chen, Hao-Chung Cheng, and Yi-Hsuan Yang. Audio prompt adapter: Unleashing music editing abilities for text-to-music with lightweight finetuning. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2024.

Fang-Duo Tsai, Shih-Lun Wu, Weijaw Lee, Sheng-Ping Yang, Bo-Rui Chen, Hao-Chung Cheng, and Yi-Hsuan Yang. MuseControlLite: Multifunctional music generation with lightweight conditioners. In *International Conference on Machine Learning (ICML)*, 2025.

Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 2006.

Apoorv Vyas, Bowen Shi, Matthew Le, Andros Tjandra, Yi-Chiao Wu, Baishan Guo, Jiemin Zhang, Xinyue Zhang, Robert Adkins, William Ngan, et al. Audiobox: Unified audio generation with natural language prompts. *arXiv preprint arXiv:2312.15821*, 2023.

Hui Wang, Yifan Yang, Zeyue Tian, Yuhang Jia, Jinghua Zhao, Long Zhou, Bing Han, Cheng Liu, Jiaming Zhou, Geng Tu, et al. UAT: Unified audio-text diffusion for audio generation, editing, and captioning. *arXiv preprint arXiv:2606.04939*, 2026.

Yashan Wang, Shangda Wu, Jianhuai Hu, Xingjian Du, Yueqi Peng, Yongxin Huang, Shuai Fan, Xiaobing Li, Feng Yu, and Maosong Sun. NotaGen: Advancing musicality in symbolic music generation with large language model training paradigms. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2025.

Yuancheng Wang, Zeqian Ju, Xu Tan, Lei He, Zhizheng Wu, Jiang Bian, et al. AUDIT: Audio editing by following instructions with latent diffusion models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Ziyu Wang, Lejun Min, and Gus Xia. Whole-song hierarchical generation of symbolic music using cascaded diffusion models. In *International Conference on Learning Representations (ICLR)*, 2024.

Minz Won, Yun-Ning Hung, and Duc Le. A foundation model for music informatics. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024.

Shih-Lun Wu, Chris Donahue, Shinji Watanabe, and Nicholas J Bryan. Music ControlNet: Multiple time-varying controls for music generation. *Transactions on Audio, Speech, and Language Processing (TASLP)*, 2024.

Shih-Lun Wu, Dave Carlton, Ryan Miyakawa, Yoon Kim, Chris Donahue, and Cheng-Zhi Anna Huang. MIDI-LLM: Improving text-to-MIDI music generation via adapting large language models. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2026a.

Shih-Lun Wu, Ge Zhu, Juan-Pablo Caceres, Cheng-Zhi Anna Huang, and Nicholas J Bryan. Stemphonic: All-at-once flexible multi-stem music generation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2026b.

Yusong Wu, Ethan Manilow, Yi Deng, Rigel Swavely, Kyle Kastner, Tim Cooijmans, Aaron Courville, Cheng-Zhi Anna Huang, and Jesse Engel. MIDI-DDSP: Detailed control of musical performance via hierarchical modeling. In *International Conference on Learning Representations (ICLR)*, 2022.

Yusong Wu, Ke Chen, Tianyu Zhang, Yuchen Hui, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, et al. Qwen3-Omni technical report. *arXiv preprint arXiv:2509.17765*, 2025.

Chenyu Yang, Shuai Wang, Hangting Chen, Wei Tan, Jianwei Yu, and Haizhou Li. SongBloom: Coherent song generation via interleaved autoregressive sketching and diffusion refinement. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yao Yao, Peike Li, Boyu Chen, and Alex Wang. JEN-1 Composer: A unified framework for high-fidelity multi-track music generation. In *AAAI Conference on Artificial Intelligence*, 2025.

Botao Yu, Peiling Lu, Rui Wang, Wei Hu, Xu Tan, Wei Ye, Shikun Zhang, Tao Qin, and Tie-Yan Liu. Museformer: Transformer with fine- and coarse-grained attention for music generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Mickaël Zehren, Marco Alunno, and Paolo Bientinesi. ADTOF: A large dataset of non-synthetic music for automatic drum transcription. In *International Society for Music Information Retrieval Conference (ISMIR)*, pp. 818–824, 2021.

Yixiao Zhang, Yukara Ikemiya, Gus Xia, Naoki Murata, Marco A Martínez-Ramírez, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. MusicMagus: Zero-shot text-to-music editing via diffusion models. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2024.

Yixiao Zhang, Yukara Ikemiya, Woosung Choi, Naoki Murata, Marco A Martínez-Ramírez, Liwei Lin, Gus Xia, Wei-Hsiang Liao, Yuki Mitsufuji, and Simon Dixon. Instruct-MusicGen: Unlocking text-to-music editing for music language models via instruction tuning. In *International Society for Music Information Retrieval Conference (ISMIR)*, 2025.

Ge Zhu, Jordan Darefsky, Fei Jiang, Anton Selitskiy, and Zhiyao Duan. Music source separation with generative flow. *IEEE Signal Processing Letters (SPL)*, 2022.

A Base-Model Pretraining and Stemphonic Baseline

In this section, we describe our base model and its fine tuned variant Stemphonic baseline. For base model, we train a general text-conditioned music generator from complete mixes. For Stemphonic, we then continue training the base generator on synchronized multitrack recordings by changing the construction of training examples and their conditioning signals.

A.1 Base-model pretraining

Latent audio representation We use the continuous tokenizer, a pretrained stereo music autoencoder (Casebeer et al., 2026), which maps 44.1 kHz stereo audio to a latent sequence $\mathbf{z}_a \in \mathbb{R}^{\mathrm{C}_a \times \mathrm{T}f}$ with $\mathrm{C}_a = 128$ channels at a frame rate of $f \approx 12.3$ Hz. The autoencoder remains fixed throughout both training stages. A $\mathrm{T} = 32$ s crop therefore yields $\mathrm{T}f = 394$ latent frames.

Rectified-flow objective Let $\mathbf{z}_a$ be the latent of a clean crop and $\boldsymbol{\epsilon}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We use the linear probability path of rectified flow (Liu et al., 2023b),

$$\mathbf{z}_a(t_a) = (1 - t_a)\,\mathbf{z}_a + t_a\,\boldsymbol{\epsilon}_a, \qquad t_a \in [0, 1], \tag{3}$$

where the endpoints $t = 0$ and $t = 1$ represent data and Gaussian noise, respectively. Given the base model's conditions $\mathcal{C}$, the DiT predicts the velocity $\hat{\mathbf{v}}_a = \mathbf{v}_\theta\big(\mathbf{z}_a(t_a), t_a, \mathcal{C}\big)$ and is trained with

$$\mathcal{L}_{\text{base}}(\theta) = \mathbb{E}_{\mathbf{z}_a \sim p_{\text{data}},\ \boldsymbol{\epsilon}_a \sim \mathcal{N}(\mathbf{0},\mathbf{I}),\ t_a \sim \mathcal{LN}} \Big\| \hat{\mathbf{v}}_a - (\mathbf{z}_a - \boldsymbol{\epsilon}_a) \Big\|_2^2, \tag{4}$$

where $\mathcal{LN}$ denotes the logit-normal timestep distribution. Generation starts from $\mathbf{z}_a(1) = \boldsymbol{\epsilon}_a$ and integrates the learned ODE to $t_a = 0$ with a first-order Euler solver.

Network and conditioning. The velocity estimator is a 1.6B diffusion transformer (DiT) (Peebles & Xie, 2023), comprising 16 blocks with hidden width 2048 and 32 attention heads. Its blocks follow the LLaMA family of transformer designs (Grattafiori et al., 2024). The flow timestep and a global audio-style representation modulate the network through adaptive normalization, while embeddings of the structured music caption are read through cross-attention in every block. The caption collects the free-form prompt together with available controls such as timing, instrumentation, vocal content, and source-quality metadata. In addition, activity conditions are concatenated to the noisy latent sequence along the feature axis.

Masked-latent context conditioning The base model is trained on fixed-length crops rather than whole compositions, so alongside the conditioners above we condition each crop on its surrounding musical material through a frame-aligned masked-latent channel. This adapts the audio-infilling paradigm of Voicebox and Audiobox (Le et al., 2023; Vyas et al., 2023) to our music latent: a single flow-matching network is trained to synthesize a masked region of the representation from the surrounding unmasked context, so that generation and editing reduce to a choice of mask at inference rather than to task-specific networks. Let $\mathbf{b} \in \{0, 1\}^{\mathrm{T}_f}$ be a binary mask selecting the frames revealed as context. Alongside the activity channel, we concatenate a masked copy of the clean latent to the noisy latent along the channel axis,

$$\tilde{\mathbf{z}}_a(t_a) = \big[\, \mathbf{z}_a(t_a);\ \mathbf{b} \odot \mathbf{z}_a \,\big], \tag{5}$$

where $[\,\cdot\,;\,\cdot\,]$ is channel-wise concatenation as in Eq. (2) and $\mathbf{b}$ is broadcast over the C_a channels, so context and target remain frame-aligned. so context and target remain frame-aligned. The revealed region is varied widely during training: from no context (a crop generated from scratch), through one-sided context (continuation), to two-sided context (in-filling between fixed endpoints), therefore, a single trained model supports autoregressive long-form generation over successive crops together with a family of editing operations such as extension, region replacement, looping, and blending, selected at inference through the mask $\mathbf{b}$.

Corpus and Training The pretraining dataset contains approximately 24,000 hours of fully licensed music. We train with AdamW (Loshchilov & Hutter, 2019) at an effective batch size of 2048 on eight NVIDIA A100 compute nodes.

A.2 Fine-tuning Specifics

Initialization and task mixture The stem model is initialized from the base-model checkpoint and fine-tuned on the licensed multitrack portion of the corpus. We preserve the grouped training and noise-sharing principle introduced by Stemphonic (Wu et al., 2026b), but apply it to the stronger base model and a unified task-conditioned pipeline. A training record contains the synchronized stems of one composition. We sample a task from *multitrack generation*, *accompaniment*, *separation*, and *removal*, and construct the target stems, aligned audio condition, task label, and conditioned-stem labels required by that task. Examples for full-mix generation are sampled separately from the master recordings. The task identity and stem labels are serialized into the same structured caption used by the base model, so all tasks share one backbone and one flow-matching objective rather than task-specific output heads.

Stem bundles and shared noise Let $\{\mathbf{z}_a^{(k)}\}_{k=1}^{K}$ be the target stems selected from one composition. Each stem is a separate batch element, but all stems in a pack share the same Gaussian noise $\boldsymbol{\epsilon}_a$, which is sampled independently across compositions. With $t_a^{(k)}$ denoting the timestep sampled for stem k, the fine-tuning paths are

$$\mathbf{z}_a^{(k)}\big(t_a^{(k)}\big) = \big(1 - t_a^{(k)}\big)\,\mathbf{z}_a^{(k)} + t_a^{(k)}\,\boldsymbol{\epsilon}_a, \qquad \boldsymbol{\epsilon}_a \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \quad k = 1, \ldots, K. \tag{6}$$

Synchronization is thus encouraged by how examples and noise are coupled; the Transformer requires neither a fixed number of stems nor an additional cross-stem attention module. The objective remains that of Eq. (4), applied to each stem.

Task-dependent conditioning For tasks that use an audio reference, the input mix or submix is encoded by the frozen autoencoder and concatenated channel-wise with each noised target latent. The corresponding tensor is set to zero when no audio context is required. Each target also receives a stem-specific caption containing the shared musical description, the requested output stem or task, and the labels of any conditioned stems. A frame-aligned activity curve can be supplied as an additional channel to control when an individual stem is active. These signals reuse the base model's conditioning adapters; the fine-tuning stage therefore changes neither the latent codec nor the Transformer backbone.

Training and inference Fine-tuning uses $32\,\mathrm{s}$ crops and the same implementation as base-model training. At inference, all stems intended for one composition are initialized with a shared Gaussian latent and paired with their respective task and stem conditions. They are evaluated as a single batch while the flow ODE is integrated from noise to data, after which the frozen decoder converts each latent trajectory endpoint to an isolated stereo waveform. This construction supports a variable number of mutually synchronized outputs in one sampling pass as well as conditional generation from an existing mix or submix.

B MIDI-VAE Reconstruction Evaluation

Joint generation operates on a continuous latent representation of the symbolic stream, so the quality of the MIDI autoencoder places an upper bound on the symbolic fidelity the generator can achieve. In this section, we verify that the MIDI-VAE reconstructs its input with negligible loss, so that errors in generated MIDI can be attributed to the generator rather than to the latent codec.

Protocol We encode each held-out MIDI crop into the VAE latent, decode it back in a single pass without iterative refinement, and compare the reconstruction against the original. The evaluation uses the checkpoint after 45k training steps. All reported counts are aggregated over the full evaluation set before computing the metrics, so longer and denser pieces contribute proportionally more.

Metrics We report precision, recall, and F1 at three granularities. *Note* compares the active pitch–time cells of the piano roll, and therefore measures how faithfully pitch content and note durations are preserved. *Onset* compares note-onset events, matched when pitch agrees and the onset time falls within a tolerance of 6 frames (i.e., 0.07 second), and therefore measures rhythmic precision. *Drum* applies the onset criterion to the percussion track, where hits are defined by instrument class and onset time rather than by duration. For each granularity, true positives (TP) are matched events, false positives (FP) are events present only in the reconstruction, and false negatives (FN) are events present only in the reference.

Results Table 11 summarizes the results. The MIDI-VAE achieves an F1 of at least 0.998 at every granularity. Note-level precision and recall are both at or above 0.999, indicating that pitch content and note durations are preserved almost exactly. Onset precision is perfect up to rounding, and the small drop in onset recall to 0.997 reflects a handful of missed onsets rather than spurious ones. Drum reconstruction is similarly accurate, with precision and recall of 0.998 and 0.999. These results indicate that the latent representation is effectively lossless for our purposes, and that the symbolic quality reported in the main experiments reflects the behavior of the joint generator.

Table 11: MIDI-VAE reconstruction performance on held-out data, under 70-millisecond tolerance.

	Precision	Recall	F1
Note	0.999	1.000	0.999
Onset	1.000	0.997	0.999
Drum	0.998	0.999	0.998

C FUSION SCHEME TRADE-OFFS

Section 3.5 introduces two ways of fusing the audio and MIDI hidden states before the DiT backbone: addition ($\mathbf{h}_{\text{add}}$) and sequence concatenation ($\mathbf{h}_{\text{cat}}$). We adopt concatenation as our default (ablated in Table 10); this section details the trade-off between the two.

Addition places the frame-wise audio and MIDI features in the same position, so the modalities interact inside every token from the first layer onward and their frame-level correspondence is enforced by construction rather than learned. A potential drawback is that information from the two modalities is superposed in one d-dimensional state: the backbone cannot attend to audio and MIDI features separately, and each output head must decode its MIDI/audio stream from the same hidden representation.

Sequence concatenation instead appends the MIDI features to the audio features along the sequence-length axis, so each modality keeps its own hidden-state positions, doubling the representational capacity compared to $\mathbf{h}_{\text{add}}$. All cross-modal information exchange can then occur through self-attention, where any audio frame can attend cleanly to any MIDI frame and vice versa. With $\mathbf{h}_{\text{cat}}$, the model needs to learn the cross-domain frame-wise correspondence, which we encourage by assigning the same RoPE (Su et al., 2024) index to every time-aligned (MIDI, audio) frame pair. We also note that $\mathbf{h}_{\text{cat}}$ incurs increased computation overhead due to the doubled sequence length.

Table 12: **Task coverage of MUSEDUO and the task-specific baselines.** One set of MUSEDUO weights spans all eleven tasks; the strongest baseline covers four. ✓ evaluated here; ○ supported, not benchmarked; (✓) restricted; ✗ outside the model's formulation. Models prefixed ○are not evaluated here; the count includes ○ but not (✓). [a] our re-implementation with expanded capabilities; [b] needs a transcription front-end for timbre transfer; [c] pitched instruments only; [d] drums only; [e] monophonic only; [f] text-prompted audio inpainting, no note control; [g] four fixed stem classes; [h] stem- not note-level; [i] symbolic output only.

	Generation						**Audio → MIDI**		**MIDI → Audio**		**Audio Edit**	
Model	Single-track (mix)	Multi-track (stem)	Text-to-MIDI	Joint MIDI + audio	Accompaniment	Separation / removal	Transcr. multi-track	Transcr. single-track	Timbre transfer	MIDI-to-audio	Note-level editing	**Tasks covered**
Text-conditioned audio generation												
MusicGen (2023)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	1
Stable Audio 3 (2026)	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	(✓)[f]	1
○ StemGen (2024)	✗	✗	✗	✗	○	✗	✗	✗	✗	✗	✗	1
○ MusicGen-Stem (2025)	○	○	✗	✗	○	✗	✗	✗	✗	✗	(✓)[h]	3
Stemphonic (2026b)	✗	✓	✗	✗	○	✗	✗	✗	✗	✗	✗	2
Stemphonic[a] (2026b)	✓	✓	✗	✗	○	✓	✗	✗	✗	✗	✗	4
Generation and separation, audio domain												
○ MSDM (2024)	✗	○[g]	✗	✗	○[g]	○[g]	✗	✗	✗	✗	✗	3
○ MSG-LD (2025)	✗	○[g]	✗	✗	○[g]	○[g]	✗	✗	✗	✗	✗	3
MGE-LDM (2025)	○	✓	✗	✗	○	✓	✗	✗	✗	✗	✗	4
Symbolic-only generation (no audio output)												
○ MuseCoco (2023)	✗	✗	○	✗	✗	✗	✗	✗	✗	✗	✗	1
Text2midi (2025)	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	1
○ Anticipatory MT (2024)	✗	✗	✗	✗	○[i]	✗	✗	✗	✗	✗	✗	1
○ Polyffusion (2023)	✗	✗	✗	✗	○[i]	✗	✗	✗	✗	✗	✗	1
MIDI-LLM (2026a)	✗	✗	✓	✗	○[i]	✗	✗	✗	✗	✗	✗	2
Transcription												
Basic Pitch (2022)	✗	✗	✗	✗	✗	✗	✗	✓[c]	✗	✗	✗	1
ADTOF (2021)	✗	✗	✗	✗	✗	✗	✗	✓[d]	✗	✗	✗	1
○ MT3 (2022)	✗	✗	✗	✗	✗	✗	○	○	✗	✗	✗	2
○ YourMT3+ (2024)	✗	✗	✗	✗	✗	✗	○	○	✗	✗	✗	2
MuScriptor (2026)	✗	✗	✗	✗	✗	✗	✓	✓	✗	✗	✗	2
MIDI-conditioned synthesis												
MIDI-DDSP[e] (2022)	✗	✗	✗	✗	✗	✗	✗	✗	✓[b]	✓	✗	2
Spectrogram Diff. (2022)	✗	✗	✗	✗	✗	✗	✗	✗	✓[b]	✓	✗	2
MuseControlLite (2025)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✓	✓	4
MUSEDUO (ours)	✓	✓	✓	✓	○	✓	✓	✓	✓	✓	✓	**11**