TOWARD INSTRUCTION-GUIDED INTERACTIVE LOW-LATENCY GENERATION FOR ACOUSTIC MUSIC

Yinghao MA (0009-0009-9500-4015) [1,*] and **Hannes GAMPER** (Hannes.Gamper@microsoft.com) (0000-0002-2148-6367) [2]

[1] *Centre for Digital Music*, **Queen Mary University of London**, London, E1 4NS UK
[2] **Microsoft Corporation**, Redmond, WA 98052 USA

ABSTRACT

Real-time, controllable music generation is important for interactive applications, yet many state-of-the-art systems remain oriented toward offline synthesis, with limited support for responsive interaction. In this work, we present *InteractiveMusicFlow* (IMF), a compact generative framework based on latent flow matching for low-latency music generation. IMF extends a text-to-audio flow matching backbone with audio-context conditioning and multi-task training, enabling a unified model that supports text-to-music generation, audio continuation, and text-instructed editing. Across MusicCaps and MusDB benchmarks, IMF demonstrates a consistent pattern: it achieves substantially lower inference latency than diffusion-based baselines, while producing outputs that remain broadly aligned with the target data distribution under standard evaluation metrics. At the same time, IMF does not yet match stronger offline models in perceptual fidelity or fine-grained control, particularly in tasks requiring precise source preservation. We view these results as evidence that flow matching provides a practical foundation for interactive music systems, where responsiveness and flexible control are essential.

1. INTRODUCTION

Recent advances in deep generative models have significantly broadened the capabilities of audio and music synthesis, supporting applications like royalty-free content generation for musicians and amatueurs. [1,2]. At the same time, sound and music computing has shifted focus beyond output quality toward the role of generative systems in creative interaction. In music contexts, interaction extends beyond one-shot text prompting to include tasks such as continuation, revision, and editing during the creative process. This reflects the inherently iterative and responsive nature of music making [2], motivating evaluation of generative models not only as content producers, but as interactive partners within a creative loop.

Despite impressive progress in music generation, many state-of-the-art (SOTA) systems remain primarily optimized for offline use. Large autoregressive models [3, 4] and latent diffusion models [5,6] can produce high-fidelity outputs, but often rely on expensive decoding, leading to noticeable inference latency. In addition, their control interfaces are commonly centered on initial prompting, while interactive use often requires a broader set of operations, such as continuing existing audio or modifying specific instrumental content, remaining a gap between generation quality and requirements of interactive workflow.

Recent work explores text-guided editing and finer control, such as Instruct-MusicGen [7] and MusicGen-Stem [8]. While improving controllability, their reliance on autoregressive backbones limits responsiveness in interactive scenarios. Moreover, to improve efficiency, non-autoregressive methods based on flow matching [9] have emerged as a promising direction. TangoFlux [10] and MeanAudio [11] demonstrate fast text-to-audio synthesis, while MusicFlow [12] extends flow-based modeling to music with unified handling of multiple tasks. However, these works focus primarily on generation quality or general audio settings, with limited attention to interactive use cases requiring both low latency and targeted control. Last, task unification has been explored in systems like UniAudio [13], which covers diverse tasks within a single model.

Recent work address real-time music generation. Magenta RT [2] and follow-up work on streaming accompaniment [14] demonstrate the feasibility of causal, token-based generation. However, autoregressive decoding remains sequential, leading to trade-offs between latency and output quality, and co-creative systems further expose challenges in responsiveness during live interaction [15]. More perspectives on live music agents highlight the need for generation paradigms that better support low-latency, interactive, and controllable musical interplay [16].

In this paper, we investigate whether a flow-matching model can unify multiple interactive music operations within a single framework. We propose *InteractiveMusicFlow* (IMF), a flow-matching model supporting multiple interaction modes: text-to-music generation, audio continuation, and text-instructed editing. It integrates a shared backbone with text and audio conditioning, and multi-task tuning, enabling diverse patterns in a unified system. We focus on demonstrating that such a design can balance between responsiveness and controllability for interactive music applications. The experiments reveal a mixed but informative outcome. IMF achieves low inference latency and supports a responsive interaction loop, while showing promising behavior in continuation and selected editing scenarios. At the same time, it does not

∗ Research conducted during an internship at Microsoft Corporation.

Copyright: © 2026. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 Unported License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

consistently match stronger baselines in perceptual fidelity, and tasks requiring precise source preservation remain challenging. We thus frame this work as an empirical study of trade-offs in interactive music generation: flow matching appears to be a viable foundation for quick responsive, multi-function systems for interactive music generation, but lacks competitiveness in an offline environment.

Our contributions are threefold. We (1) present IMF, a versatile flow-matching framework for text-to-music generation, audio continuation, and text-instructed editing; (2) demonstrate it can operate under a low-latency regime suitable for interactive use; and (3) provide an empirical analysis highlighting the strengths and limitations of IMF.

2. METHODOLOGY

As illustrated in Figure 1, we present IMF, a compact latent rectified-flow model for low-latency, controllable music generation, which support generation, continuation and editing within a single model. Here, real-time refers to generating short audio segments fast enough to support a responsive interaction loop.

2.1 Unified Interactive Task Formulation

We formulate all tasks in this paper as conditional generation of a target 10-second music segment. Let $X \in \mathbb{R}^{2\times 441000}$ denote a stereo waveform and let Z be its latent representation from the frozen VAE of Stable Audio Open [17]. Our model learns a conditional generator

$$\hat{Z} = f_\theta(z_t, t, c), \tag{1}$$

where z_t is the noised latent at time t, and c denotes the conditioning information. The task-specific difference lies only in how c is constructed during interplaying.

For **text-to-music generation**, c contains a text description and a target duration. For **audio continuation**, c additionally includes the latent representation of a preceding 10-second audio prompt, which provides local musical context for generating the next segment. For **text-instructed editing**, c contains both a source audio segment and an instruction such as adding, removing, or isolating an instrument. Under this view, the three tasks share the same output space and the same latent generator; only the available control signals differ. This unified formulation is intended to make the interactive setting explicit and to avoid training separate models for each operation.

IMF uses three conditioning sources. First, for **text conditioning**, we encode the input text using a pretrained FLAN-T5 encoder [18], producing a sequence of token embeddings E_{text} and a pooled text embedding c_{text}. Second, for **audio-context conditioning**, an optional preceding 10-second prompt X_{prompt} is encoded by the same frozen VAE into a latent sequence Z_{prompt}. This prompt latent is then projected into the model conditioning space and used to represent prior musical context. Third, for **duration conditioning**, a scalar target duration is mapped to an embedding c_{dur} by a small neural network.

A key design choice is to use the audio condition at two levels. We derive a *global* context vector from the prompt audio and fuse it with the pooled text embedding to provide high-level control over the generation. In parallel, we inject the prompt audio as *token-level* conditioning through cross-attention, so that the model can access more detailed local context during generation. Intuitively, the text specifies *what to generate or modify*, while the audio prompt specifies *what musical context to continue or preserve*. This simple framing allows the backbone to support both prompt-only generation and context-dependent interaction.

2.2 Model Architecture

We use a FluxTransformer composed of an initial stack of Multimodal Diffusion Transformer (MMDiT) blocks [19], followed by Diffusion Transformer (DiT) [20]. Such design provides a practical compromise between multimodal conditioning capacity and parameter efficiency. In our setting, the transformer backbone keep the same hyperparameters as Tangoflux's. Furthermore, IMF extends the Tangoflux transformer backbone by allowing the model to attend to the latent representation of a preceding 10-second audio segment during generation. This additional cross-attention branch is intended to support tasks that depend on existing musical material, especially continuation and instruction-guided editing.

2.3 Flow Matching

IMF is trained with the same latent rectified-flow formulation as [10]. Let z_1 denote the target audio latent and let $z_0 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ be a Gaussian prior sample. For a sampled time $t \in [0, 1]$, we define the straight-line interpolation

$$z_t = tz_1 + (1-t)z_0. \tag{2}$$

The corresponding target velocity is

$$v^\star = z_1 - z_0. \tag{3}$$

The model $v_\theta(z_t, t, c)$ is trained to predict this velocity conditioned on the noised latent z_t, the time step t, and the conditioning variables c. We optimize the standard flow-matching regression loss

$$\mathcal{L}_{\text{FM}} = \mathbb{E}_{z_1, z_0, t, c}\left[|0v_\theta(z_t, t, c) - (z_1 - z_0)|\, 0_2^2\right]. \tag{4}$$

At inference time, we sample z_0 from the Gaussian prior and integrate the learned velocity field with an ODE solver to obtain the final latent $\hat{z}_1$, which is then decoded by the frozen VAE. We use a small number of solver steps, which is one reason this formulation is attractive for low-latency generation. While keeping a competitive potential of autoregressive or diffusion-based generation in terms of audio quality, flow-based model offers a more reasonable inference speed for interactive generation.

3. EXPERIMENTS

We evaluate IMF on three interactive music generation settings: text-to-music generation, audio continuation, and text-instructed editing such as adding, removing, or extracting instruments. In this section, we clarify multi-task data curation, training receipes, and evaluation metrics.

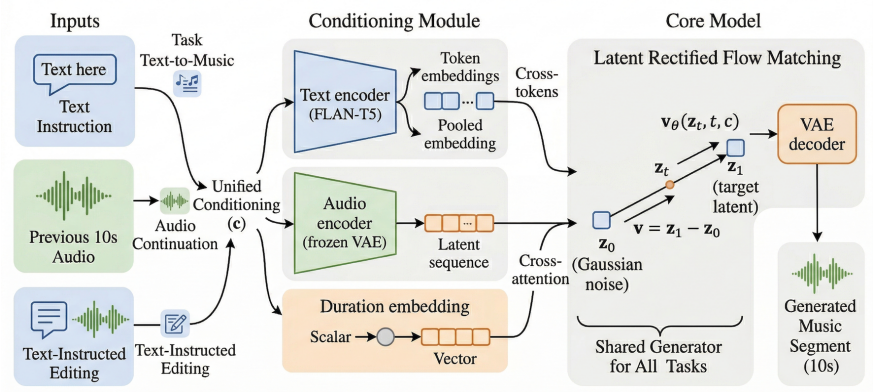


Figure 1. InteractiveMusicFlow (IMF), a compact model supporting multiple interactive music generation tasks, featuring unified conditioning and efficient generation via latent rectified flow matching..

3.1 Datasets

Our training data combine large-scale corpora for general generation with source-separated music data for continuation and editing. All audio is resampled to 44.1kHz stereo and segmented into non-overlapping 10-second clips. For a track of length T, we extract $\lfloor T/10 \rfloor$ clips for text-to-music training. For continuation, adjacent clips are paired sequentially (e.g., a 35-second track yields three clips and two continuation pairs). This fixed segmentation ensures a simple and consistent training pipeline across datasets.

We use three datasets for training: **MTG-Jamendo** [21], **FMA** [22], and **Freesound Dataset** [23], which provide broad musical coverage and diverse metadata. Rather than relying on free-form captions, we curate textual annotations from metadata fields such as instruments, genre, mood/emotion, and available descriptions. These serve as lightweight text supervision to expose the model to diverse music–text correspondences. For FMA, we exclude tracks with the `experimental` tag to reduce distributional instability. We incorporate **MusicCaps** [4], which provides music clips paired with detailed human-written descriptions. It serves both as a music–text resource and as a benchmark-aligned dataset for text-to-music and continuation. For continuation, the preceding 10-second segment is used as the audio prompt and the subsequent segment as the target. When text annotations are available, they are used as auxiliary conditioning; otherwise, the model relies primarily on audio context.

For text-instructed editing, we construct supervised training data from **MusDB18-HQ** [24], which provides mixtures and four stems (*vocals*, *bass*, *drums*, *other*). We use the official train split for training and the test split for evaluation, and derive three editing tasks from these stems:

- **Adding:** We simulate editing by mapping a source mixture with fewer stems to a target with one additional stem (from 1-3 stems to a richer mixture).
- **Removing:** We simulate subtractive editing by removing a specified stem from the source mixture.
- **Extracting:** We construct extraction pairs where the input is full mixture and target is a selected stem.

We generate all valid segment-level pairs from the training split. Instructions are created using simple templates with add, remove, and extract, combined with instrument names (e.g., add drums, remove vocals, extract bass), ensuring clarity and reproducibility. For evaluation, we use the MusDB18-HQ test split for editing tasks, and the official evaluation split of MusicCaps [4] for text-to-music generation and audio continuation.

3.2 Training Recipe

Training is performed in two stages. In both stages, audio is processed as 10-second stereo clips at 44.1 kHz, and the model operates in the latent space of the frozen VAE. We train on 8 NVIDIA A100 40GB GPUs with a per-device batch size of 14, giving a total batch size of 112.

Stage 1: general pre-training. We first train the model on the combined MTG-Jamendo, FMA, and Freesound corpora to learn a general music/audio generation prior from text and metadata supervision. This stage is intended to provide broad coverage of musical content before introducing task-specific interaction data.

Stage 2: multi-task fine-tuning. We then fine-tune the same model on the combined multi-task mixture including the general music–text data, MusicCaps, and MusDB18-HQ editing pairs. This stage exposes the model to three interaction modes within a shared conditional generation framework: text-to-music, continuation, and editing. In this paper, the term *multi-task instructions* therefore refers to the textual control signals used across these tasks, including descriptive metadata for generation and manually templated action-oriented instructions for editing.

For optimization, we use AdamW with a peak learning rate of 5×10^{-4}, linear warmup for the first 1,500 steps, and cosine decay thereafter. The general pre-training stage runs for 120,000 steps. The multi-task fine-tuning stage uses the same optimizer and scheduler. Examples from different tasks are mixed during fine-tuning so that the model learns all task modes within a single backbone.

3.3 Evaluation Metrics and Baselines

We evaluate the performance using a set of standard objective metrics and compare against relevant SOTA baselines.

To assess the fidelity and realism of the generated audio, we use **Fréchet Audio Distance** (FAD) (↓), which

Model	FAD ↓	KL ↓	FD (PANN) ↓	CLAP ↑
MusicGen (1.5B) [3]	3.40	1.23	24.1	0.37
UNIAUDIO (1B) [13]	3.65	1.90	-	-
AudioLDM-2 (746M) [5]	3.13	1.20	18.8	0.40
MusicFlow (546M) [12]	**2.69**	1.23	13.2	**0.52**
Ours (IMF, 515M)	5.15	**0.72**	**1.91**	0.20

Table 1. Objective evaluation for Text-to-Music Generation on the MusicCaps (MC) test set.

measures perceptual similarity between the distributions of generated and real audio embeddings [25]; **KL-Divergence** (↓), which compares the label distributions of generated and real audio using a pre-trained audio classifier; **FD (PANN)** (↓), a variant of FAD that utilizes a powerful PANN audio event recognition model as the feature extractor [26]. All are under the default setting of `audioldm_eval` toolkit [6] Following recent work on FAD evaluation [27], we report **FAD reference** (↓) with multiple reference sets, as the choice of reference distribution can significantly influence the resulting scores.

Besides, **CLAP score** (↑) measures the cosine similarity between text and audio embeddings to evaluate text-audio correspondence [28].

Furthermore, **Real-Time Factor (RTF)** (↓) is defined as the ratio between the audio generation time and the duration of the generated audio to quantify generation latency.

4. RESULTS AND ANALYSIS

We evaluate IMF across three interactive music tasks: text-to-music generation, audio continuation, and text-instructed editing, assessing whether a single compact flow-matching model can support multiple interaction modes under a low-latency setting.

4.1 Text-to-Music Generation

Table 1 reports results on the MusicCaps test set. IMF does not surpass the strongest baselines on perceptual fidelity or text-audio embedding alignment, as reflected by higher FAD and lower CLAP scores compared to diffusion-based and flow-based models. This gap is expected given that IMF is trained under a compact, low-latency regime without additional preference optimization stages. However, IMF achieves substantially lower KL divergence and FD(PANN) values. These results indicate that the generated samples remain close to the target distribution under classifier- and embedding-based statistics. While these metrics do not directly imply improved musical structure or fine-grained semantic alignment, they suggest that IMF can produce broadly appropriate conditioned music within a constrained inference budget.

Table 2 further examines the trade-off between generation quality and inference efficiency under different FAD reference sets. Consistent with the results in Table 1, IMF does not outperform larger or diffusion-based models in terms of FAD across all reference distributions, particularly when evaluated against ground-truth (GT) references. However, IMF achieves comparable FAD to MusicGen under the FMA reference, suggesting that the generated samples remain reasonably close to real-world music distributions under certain evaluation conditions. More importantly, IMF achieves a substantially lower RTF, corresponding to approximately 7× faster inference than MusicGen. These results highlight a consistent trade-off: IMF sacrifices some perceptual fidelity relative to stronger offline models, while providing significantly improved efficiency. Therefore, we interpret this table as supporting the practical advantage of IMF in low-latency settings.

Model	FAD reference ↓			RTF ↓
	fma-pop	MC	musCC	
MusicGen (1.5B) [3]	284.7	**876.5**	441.5	1.8
MusicGen (3.3B) [3]	**265.6**	880.0	**418.5**	1.8
AudioLDM-2 (746M) [5]	303.0	887.4	470.0	1.4
Ours (IMF, 515M)	267.9	998.2	468.8	**0.2**

Table 2. Inference on 500 MC captions. FAD using CLAP 2023 and three reference sets [27]. MusicGen leads on quality, while IMF has 7× faster single-sample inference

Model	#Param	RTF ↓	FAD ↓	KL ↓	FD ↓	CLAP ↑
AudioLDM-2 [5]	746M	1.5	2.08	2.80	25.1	0.53
MusicFlow [12]	546M	-	**1.63**	0.49	6.50	**0.56**
Ours (IMF)	515M	**0.2**	2.38	**0.57**	**1.74**	0.31

Table 3. Objective evaluation for Music Continuation.

4.2 Audio Continuation

Table 3 presents results for audio continuation. IMF achieves the lowest real-time factor, indicating a clear advantage in inference speed. This supports the design goal of enabling short-turn interactive generation. On quality-oriented metrics, IMF remains behind MusicFlow, particularly on FAD and CLAP. However, IMF shows competitive KL and notably low FD(PANN) values. These results suggest that the generated continuations are close to the target distribution in the evaluated feature space.

Importantly, these metrics do not directly measure continuation-specific properties such as beat alignment, melodic continuity, or timbre consistency. Therefore, we interpret the continuation results as encouraging but incomplete evidence. In practice, we observe that IMF often preserves high-level musical attributes such as tempo and general style, while exhibiting variability in timbral details.

4.3 Text-Instructed Editing

Table 4 summarizes results for text-instructed editing on MusDB18-HQ. We analyze performance separately for adding, removing, and isolating instruments. For Adding instruments, IMF achieves comparable or slightly improved KL and FD(PANN) values relative to InstructMusicGen, while remaining behind in FAD and CLAP. This suggests that the model can incorporate requested changes at a distributional level, although the perceptual fidelity of the edited output is still lower than that of specialized baselines. Instrument removal is the most challenging setting for IMF. The model underperforms the baseline across all reported metrics, indicating that selectively subtracting a source while preserving the remaining mixture is not yet handled reliably. This result is consistent with the difficulty of source disentanglement in a unified generative

Model	FAD ↓	KL ↓	FD (PANN) ↓	CLAP ↑
Adding Instruments				
Instruct-MusicGen [7]	**2.70**	0.84	6.72	**0.22**
Ours (IMF)	6.11	**0.83**	**6.13**	0.19
Removing Instruments				
Instruct-MusicGen	**2.06**	**0.25**	**2.08**	**0.24**
Ours (IMF)	2.64	0.71	4.23	0.19
Isolating Instruments				
Instruct-MusicGen	4.92	**1.15**	16.68	0.14
Ours (IMF)	**2.62**	1.31	**11.99**	**0.17**

Table 4. Objective evaluation for Text-Instructed Editing on the MusDB18-HQ test set.

model. In contrast, IMF performs strongly in the isolation setting, achieving better FAD and FD(PANN) scores than the baseline and competitive CLAP scores. This suggests that generating a target source in isolation is more tractable than removal-based editing, which requires preserving all other components. Overall, these results highlight that editing performance varies significantly by operation type. IMF shows promise for additive and extraction tasks, while removal remains an open challenge.

4.4 Real-time Analysis and Application

Latency is a central consideration in this work. IMF achieves an real time factor (RTF) of 0.2, generating a 10-second audio segment in under 2 seconds on NVIDIA A100, the same hardware with baselines. This is substantially faster than typical diffusion-based generation pipelines and enables a responsive interaction loop for short audio segments. We do not interpret this as fully real-time generation in the strict causal sense. However, it suggests the iterative workflows where users generate, listen, and modify short segments are practically feasible.

4.5 Qualitative Analysis and Limitations

Qualitative analysis provides additional insight into the model's behavior beyond the reported metrics. In audio continuation, IMF often preserves global musical attributes such as tempo, rhythmic pattern, and general genre. However, we observe consistent timbral drift: for example, a continued guitar phrase may retain its role but change tone or production characteristics, suggesting audio conditioning captures high-level context but not low-level acoustic details. In removal editing scenarios, IMF may suppress not only the target instrument but also related components, indicating precise source disentanglement is not yet achieved. In contrast, isolation tasks are more stable. These observations suggest that improving timbre preservation and source-aware control remains an important direction for future work.

5. CONCLUSION AND FUTURE WORK

We presented *InteractiveMusicFlow*, a compact latent flow-matching model for low-latency music generation across three interaction modes: text-to-music, audio continuation, and text-instructed editing. Results show that a single flow-based backbone can support these tasks with favorable inference speed, enabling a unified and responsive framework. However, we observe limitations in perceptual fidelity, timbre consistency, and source-preserving editing, particularly for removal operations. We therefore position this work as an exploratory step toward interactive music generation rather than a replacement for high-quality offline systems, highlighting a key trade-off between responsiveness and fine-grained control. Future work will focus on improving conditioning mechanisms for better timbre preservation and continuation consistency, developing task-specific evaluation protocols including human studies, and extending the framework to richer interaction modalities and longer temporal contexts. We hope this work provides a useful reference for designing interactive music systems that balance quality, controllability, and responsiveness, and benefits DAW-based iterative generation, live performance, and etc.

Acknowledgments

Yinghao Ma is a research student at UKRI Centre for Doctoral Training in AI&Music, supported by UK Research & Innovation [grant number EP/S022694/1]. He also thank Dr. Emmanouil Benetos & Yisu Zong for proofreading.

6. REFERENCES

[1] Y. Ma, A. Øland, A. Ragni, B. M. Del Sette, C. Saitis, C. Donahue, C. Lin, C. Plachouras, E. Benetos, E. Shatri *et al.*, "Foundation models for music: A survey," *arXiv preprint arXiv:2408.14340*, 2024.

[2] L. Team, A. Caillon, B. McWilliams, C. Tarakajian, I. Simon, I. Manco, J. Engel, N. Constant, P. Li, T. I. Denk *et al.*, "Live music models," *arXiv preprint arXiv:2508.04651*, 2025.

[3] J. Copet, F. Kreuk, I. Gat, T. Remez, D. Kant, G. Synnaeve, Y. Adi, and A. Défossez, "Simple and controllable music generation," *Advances in Neural Information Processing Systems*, vol. 36, pp. 47 704–47 720, 2023.

[4] A. Agostinelli, T. I. Denk, Z. Borsos, J. Engel, M. Verzetti, A. Caillon, Q. Huang, A. Jansen, A. Roberts, M. Tagliasacchi *et al.*, "Musiclm: Generating music from text," *arXiv preprint arXiv:2301.11325*, 2023.

[5] H. Liu, Y. Yuan, X. Liu, X. Mei, Q. Kong, Q. Tian, Y. Wang, W. Wang, Y. Wang, and M. D. Plumbley, "Audioldm 2: Learning holistic audio generation with self-supervised pretraining," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 32, pp. 2871–2883, 2024.

[6] H. Liu, Z. Chen, Y. Yuan, X. Mei, X. Liu, D. Mandic, W. Wang, and M. D. Plumbley, "Audioldm: Text-to-audio generation with latent diffusion models," in *International Conference on Machine Learning*. PMLR, 2023, pp. 21 450–21 474.

[7] Y. Zhang, Y. Ikemiya, W. Choi, N. Murata, M. A. M. Ramírez, L. Lin, G. Xia, W. Liao, Y. Mitsufuji,

and S. Dixon, "Instruct-musicgen: Unlocking text-to-music editing for music language models via instruction tuning," in *Proceedings of the 26th International Society for Music Information Retrieval Conference, ISMIR 2025, Daejeon, South Korea, September 21-25, 2025*, 2025, pp. 328–336.

[8] S. Rouard, R. San Roman, Y. Adi, and A. Roebel, "Musicgen-stem: Multi-stem music generation and edition through autoregressive modeling," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[9] Y. Lipman, R. T. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *11th International Conference on Learning Representations, ICLR 2023*, 2023.

[10] C.-Y. Hung, N. Majumder, Z. Kong, A. Mehrish, A. A. Bagherzadeh, C. Li, R. Valle, B. Catanzaro, and S. Poria, "Tangoflux: Super fast and faithful text to audio generation with flow matching and clap-ranked preference optimization," *arXiv preprint arXiv:2412.21037*, 2024.

[11] X. Li, J. Liu, Y. Liang, Z. Niu, W. Chen, and X. Chen, "Meanaudio: Fast and faithful text-to-audio generation with mean flows," *arXiv preprint arXiv:2508.06098*, 2025.

[12] K. Prajwal, B. Shi, M. Lee, A. Vyas, A. Tjandra, M. Luthra, B. Guo, H. Wang, T. Afouras, D. Kant *et al.*, "Musicflow: cascaded flow matching for text guided music generation," in *Proceedings of the 41st International Conference on Machine Learning*, 2024, pp. 41 052–41 063.

[13] D. Yang, J. Tian, X. Tan, R. Huang, S. Liu, X. Chang, J. Shi, S. Zhao, J. Bian, X. Wu *et al.*, "Uniaudio: An audio foundation model toward universal audio generation," *arXiv preprint arXiv:2310.00704*, 2023.

[14] Y. Wu, M. Wang, H. Lei, S. Brade, L. Blanchard, S.-L. Wu, A. Courville, and A. Huang, "Streaming generation for music accompaniment," *arXiv preprint arXiv:2510.22105*, 2025.

[15] L. Bradshaw, A. Spangher, S. Biderman, and S. Colton, "The ghost in the keys: A disklavier demo for human-ai musical co-creativity," *CoRR*, vol. abs/2511.01663, 2025.

[16] Y. Kim, S. Brade, A. Wang, D. Zhou, H. Kim, B. Wang, S.-J. Lee, H. F. F. Garcia, C.-Z. A. Huang, and C. Donahue, "A design space for live music agents," *arXiv preprint arXiv:2602.05064*, 2026.

[17] Z. Evans, J. D. Parker, C. Carr, Z. Zukowski, J. Taylor, and J. Pons, "Stable audio open," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2025, pp. 1–5.

[18] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma *et al.*, "Scaling instruction-finetuned language models," *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.

[19] P. Esser, S. Kulal, A. Blattmann, R. Entezari, J. Müller, H. Saini, Y. Levi, D. Lorenz, A. Sauer, F. Boesel *et al.*, "Scaling rectified flow transformers for high-resolution image synthesis," in *International Conference on Machine Learning*. PMLR, 2024, pp. 12 606–12 633.

[20] W. Peebles and S. Xie, "Scalable diffusion models with transformers," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2023, pp. 4195–4205.

[21] D. Bogdanov, M. Won, P. Tovstogan, A. Porter, and X. Serra, "The mtg-jamendo dataset for automatic music tagging." ICML, 2019.

[22] M. Defferrard, K. Benzi, P. Vandergheynst, and X. Bresson, "FMA: A dataset for music analysis," in *Proceedings of the 18th International Society for Music Information Retrieval Conference, ISMIR 2017, Suzhou, China, October 23-27, 2017*, S. J. Cunningham, Z. Duan, X. Hu, and D. Turnbull, Eds., 2017, pp. 316–323.

[23] E. Fonseca, J. Pons, X. Favory, F. Font, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra, "Freesound datasets: A platform for the creation of open audio datasets." in *ISMIR*, 2017, pp. 486–493.

[24] Z. Rafii, A. Liutkus, F.-R. Stöter, S. I. Mimilakis, and R. Bittner, "MUSDB18-HQ-an uncompressed version of MUSDB18," 2019.

[25] K. Kilgour, M. Zuluaga, D. Roblek, and M. Sharifi, "Fréchet audio distance: A reference-free metric for evaluating music enhancement algorithms," in *Proc. Interspeech 2019*, 2019, pp. 2350–2354.

[26] Q. Kong, Y. Cao, T. Iqbal, Y. Wang, W. Wang, and M. D. Plumbley, "Panns: Large-scale pretrained audio neural networks for audio pattern recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2880–2894, 2020.

[27] A. Gui, H. Gamper, S. Braun, and D. Emmanouilidou, "Adapting frechet audio distance for generative music evaluation," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2024, Seoul, Republic of Korea, April 14-19, 2024*. IEEE, 2024, pp. 1331–1335. [Online]. Available: https://doi.org/10.1109/ICASSP48485.2024.10446663

[28] Y. Wu, K. Chen, T. Zhang, Y. Hui, T. Berg-Kirkpatrick, and S. Dubnov, "Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation," in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.